

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Měřicí karta s Bluetooth low energy
Measuring Card with Bluetooth Low Energy

Zadání bakalářské práce

Student: **Michal Strýček**
Studijní program: B2649 Elektrotechnika
Studijní obor: 3901R039 Biomedicínský technik
Téma: **Měřicí karta s Bluetooth low energy**
Measuring Card with Bluetooth Low Energy
Jazyk vypracování: čeština

Zásady pro vypracování:

1. Seznámení se s problematikou bluetooth low energy.
2. Návrh a realizace firmware pro zvolený bluetooth LE modul pro měření alespoň 3 analogových signálů a zpracování alespoň jednoho digitálního (SPI nebo I2C) a přenos do mobilního smart zařízení.
3. Návrh a realizace software pro nastavení parametrů měření pomocí zvoleného Bluetooth LE modulu a vizualizace měřených dat.
4. Provedení testovacích měření
5. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] *Body Sensor Network*. Editor Guang-Zhong YANG. London (UK): Springer-Verlag, 2006. 493 s. ISBN 978-1-84628-272-0.
[2] ČERNÝ, Martin a Marek PENHAKEK. *Biotelemetrie*. 1.vyd. Ostrava: VŠB – TU Ostrava, 2007. ISBN 978-80-248-1605-0.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Martin Černý, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018


doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

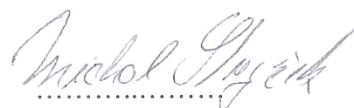



prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 27. dubna 2018


.....
podpis studenta

Poděkování

Především bych touto cestou rád poděkoval vedoucímu *doc. Ing. Martinovi Černému, Ph.D.* za odborné vedení, cenné rady a věnovaný čas, při vytváření této bakalářské práce.

Abstrakt

Tato bakalářská práce pojednává o návrhu měřicí karty s Bluetooth Low Energy. V úvodu práce je rozebrána problematika týkající se Bluetooth Low Energy a volba Bluetooth modulu. Následně je popsána struktura navrženého GATT pro komunikaci mezi BGM113 a chytrým zařízením s operačním systémem Android. Poté je předveden firmware pro BGM113, ten umožňuje převod analogových signálů a měření teploty a relativní vlhkosti pomocí digitálního senzoru. Dále je prezentována vytvořená aplikace pro Android, pomocí které dochází k vizualizaci dat, nastavení parametrů měření a exportu naměřených dat. V závěru jsou uvedeny naměřené výsledky, energetická náročnost řešení a grafické uživatelské rozhraní aplikace.

Klíčová slova

Bluetooth Low Energy, BGM113, firmware, Android, GATT, měřicí karta

Abstract

This bachelor thesis deals with designing a Bluetooth Low Energy Measuring Card. First part of this thesis describes the concept of Bluetooth Low Energy and selection of the Bluetooth module. Afterwards the structure of designed GATT for communication between BGM113 module and Android smart device is described. Subsequently the firmware for BGM113 module is introduced which enables measurement of temperature and relative humidity using digital sensor, furthermore analog to digital conversion is made possible. After that the Android application is presented. This application allows users to visualize and export data and set up measurement parameters. Lastly measurement results and energy profiles are presented, along with the graphical users interface of the application.

Keywords

Bluetooth Low Energy, BGM113, firmware, Android, GATT, measuring card

Obsah

Seznam použitých symbolů a zkratk.....	6
Seznam ilustrací.....	7
Seznam tabulek.....	8
Úvod.....	9
1 Teoretická část.....	10
1.1 Bluetooth low energy	10
1.1.1 Kompatibilita.....	11
1.1.2 Architektura technologie	12
1.2 Výběr Bluetooth LE modulu	15
1.3 Blue Gecko Module Wireless Starter Kit.....	16
2 Praktická část	17
2.1 GATT	17
2.1.1 Generic Access a Device Information.....	18
2.1.2 Environmental Sensing	19
2.1.3 Automation IO	20
2.2 Firmware BGM113.....	23
2.2.1 Struktura projektu.....	23
2.2.2 Volba Event Listener.....	24
2.2.3 Třídění příchozích událostí a jejich struktura	24
2.2.4 Funkce pro využití digitálního senzoru Si 7021	28
2.2.5 Funkce pro A/D převodník.....	29
2.3 Návrh aplikace pro Android	30
2.3.1 Návrh základní struktury aplikace.....	31
2.3.2 Hlavní aktivita	32
2.3.3 Aktivita Measure	34
2.3.4 Fragment tempRh.....	38
2.3.5 Fragment Analog.....	39
2.3.6 Fragment Settings.....	40
2.4 Výsledky měření a prezentace uživatelského rozhraní.....	41
2.4.1 Uživatelské rozhraní Android	41
2.4.2 Naměřené analogové signály.....	45
2.4.3 Naměřené enviromentální veličiny.....	47
2.4.4 Energetické Profily BGM113.....	48
Závěr.....	52
Literatura.....	54
Seznam příloh	55

Seznam použitých symbolů a zkratk

hex	Hexadecimální
API	Rozhraní pro programování aplikací (Application Programming Interface)
ATT	Attribute Protocol
A/D převodník	Analogově digitální převodník
Bluetooth SIG	Bluetooth Special Interest Group
Bluetooth LE	Bluetooth Low Energy
EKG	Elektrokardiogram
GAP	Generic Access Profile
GATT	Generic Attribute Protocol
HCI	Host Controller Interface
I ² C	Inter—Integrated Circuit
ID	Identifikátor (Identifier)
L2CAP	Logical Link and Adaptation Protocol
MCU	Mikrokontroler (Microcontroller unit)
MSB	Nejvýznamnější bit (Most Significant Bit)
PPG	Fotoplethysmograf (Photoplethysmography)
RAM	Random Access Memory
SM	Security Manager
USB	Universální sériová sběrnice (Universal Serial Bus)
UTF-8	Unicode Transformation Format
UUID	Universally unique identifier

Seznam ilustrací

Obrázek 1: Zobrazení pásma a kanálů Bluetooth, spolu s interferencemi Wi-Fi sítí.....	11
Obrázek 2: Přehled architektury Bluetooth LE.....	12
Obrázek 3: Proces výměny dat mezi serverem a klientem.....	12
Obrázek 4: Role zařízení.....	13
Obrázek 5: Komunikace s Acknowledgement a bez. Převzato z [1]	14
Obrázek 6: Struktura L2CAP paketu. Převzato z [1]	15
Obrázek 7: Přehled dostupného hardware. Převzato z [2]	16
Obrázek 8: Struktura navrženého GATT.....	17
Obrázek 9: Struktura služeb Generic Access a Device Information.....	18
Obrázek 10: Struktura služby Environmental Sensing.....	19
Obrázek 11: Struktura služby Automation IO.....	20
Obrázek 12: Struktura upravené charakteristiky Analog.....	20
Obrázek 13: Příklad bajtů pro volbu pinu pro A/D převodník.....	22
Obrázek 14: Struktura charakteristiky User Control Point.....	22
Obrázek 15: Ovládání časovačů měření.....	25
Obrázek 16: Switch pro třídění události změny hodnoty.....	26
Obrázek 17: Struktura case pro časovače.....	27
Obrázek 18: Vývojový diagram funkce pro měření teploty a vlhkosti.....	28
Obrázek 19: Funkce adcSingleScan().	29
Obrázek 20: Struktura funkce režimu Override.....	30
Obrázek 21: Struktura navržené aplikace.....	31
Obrázek 22: Diagram užití hlavní aktivity.....	32
Obrázek 23: Průběh metody při nalezení výsledků.....	33
Obrázek 24: Sled metod pro vytvoření nového spojení.....	36
Obrázek 25: Průběh metody pro volbu charakteristiky.....	36
Obrázek 26: Zpracování příchozí notifikace teploty/vlhkosti.....	37
Obrázek 27: Průběh zpracování hodnot z charakteristiky Analog.....	38
Obrázek 28: Diagram užití fragmentu Settings.....	40
Obrázek 29: Hlavní aktivita, Bluetooth LE skener.....	41
Obrázek 30: Obrazovka s fragmentem tempRh.....	42
Obrázek 31: Zobrazení pracujícího fragmentu analog.....	43
Obrázek 32: Další část fragmentu Settings.....	44
Obrázek 33: Uživatelské rozhraní fragmentu Settings.....	44
Obrázek 34: Naměřené EKG.....	45
Obrázek 35: Měření zvolených analogových singálů.....	46
Obrázek 36: Naměřená relativní vlhkost.....	47
Obrázek 37: Naměřená teplota.....	47
Obrázek 38: Energetický profil průběhu Advertisingu.....	48
Obrázek 39: Profil připojení za klidu.....	49
Obrázek 40: Profil měření teploty a relativní vlhkosti.....	49
Obrázek 41: Profil měření analogového signálu při vzorkovací frekvenci 100 Hz.....	50
Obrázek 42: Režim Override, vzorkovací frekvence 200 Hz interval spojení rychlý.....	50

Seznam tabulek

Tabulka 1: Příklad navyšování rychlosti přenosu novějších technologií. [1]	10
Tabulka 2: Kompatibilita mezi standardy bluetooth. [1]	11
Tabulka 3: Hlavní rozdíly mezi zvolenými Bluetooth Smart Moduly	15
Tabulka 4: Specifikace spojení BGM113 a tabletu Google Nexus 7.....	21
Tabulka 5: UUID potřebné ke komunikaci, uvedeny v hex formě.	35
Tabulka 6: Hodnoty využité pro nastavení User Control Point.	39
Tabulka 7: Porovnání výsledků měření, doba měření přibližně 60 s.	51

Úvod

Bluetooth Low Energy je technologie pro bezdrátovou komunikaci na krátké vzdálenosti ubírající se opačným směrem než většina digitálního světa. Namísto snahy o nejrychleji dostupná data se soustředí na uskutečnění komunikace za co nejnižších energetických nároků. V dnešní době, kdy musí být chytré i hodinky jsou technologie spojující je s okolím v zajímavém postavení. Propojení senzorů s chytrým zařízením se využívá jak v telemedicině, tak ve sportovní medicíně. V době, kdy se objevují nové možnosti jako Bluetooth Mesh Networking, umožňující vytvořit systém spolupracujících jednotek, je tato problematika o to aktuálnější. Možnost podílet se na návrhu chytré cyklistické přilby mne proto oslovila. Po vytyčení okruhů, kterými se bylo potřeba v rámci projektu zabývat, jsem si vybral realizaci měřicí karty využívající Bluetooth Low Energy.

První část této práce se věnuje teoretickému úvodu do Bluetooth Low Energy, probírá architekturu technologie a základní principy přenosu informací. V druhé části je postupně popsána navržená komunikační kostra a její části. Následuje seznámení s vytvořeným firmware pro Bluetooth LE modul BGM113 a problematiky s ním spojené. Dále se práce zabývá sestavením aplikace pro systém Android a dílčími problémy. Závěrečná část prezentuje energetickou náročnost navrženého firmware a zhodnocuje kvalitu naměřených dat.

1 Teoretická část

1.1 Bluetooth low energy

Bluetooth low energy (dále Bluetooth LE) bylo navrženo jako doplňková část klasickému Bluetooth. Zároveň také jako samostatná bezdrátová technologie s nejnižší spotřebou energie, kterou je možné navrhnout a sestavit.

Klasický Bluetooth byl zpočátku užíván k přenosu hudby. S postupem času a s nárůstem různých odvětví, ve kterých byl používán, rostl požadavek na jeho maximální přenosovou rychlost.

Oproti tomu Bluetooth LE právě z důvodu snižování spotřeby limituje rychlost přenosu dat a prodlužuje komunikační intervaly. Nepodporuje tak na rozdíl od jiných technologií vysokou přenosovou rychlost, ale spíše zaměření na úsporu energie. Zařízení tak mohou fungovat při napájení knoflíkovou baterií i po dobu několika let. Neznamená to ovšem, že zařízení Bluetooth LE, mají malý dosah signálu, právě naopak, většina má dosah větší než klasický Bluetooth.

[1]

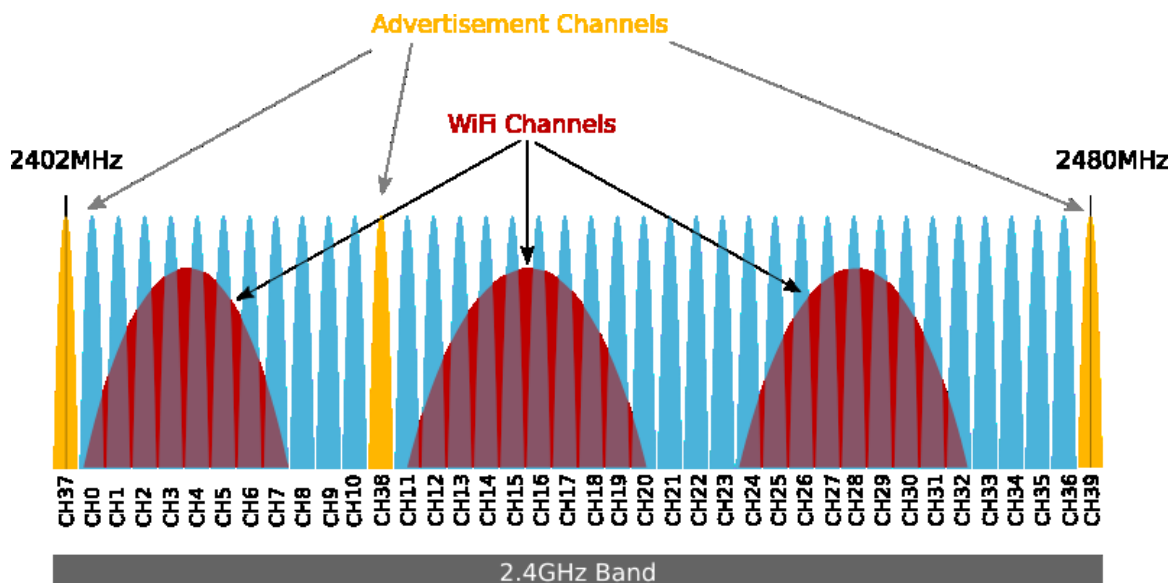
Tabulka 1: Příklad navyšování rychlosti přenosu novějších technologií. [1]

Wi-Fi	Bluetooth
802.11: 2Mbps	v1.1: 1Mbps
802.11b: 11Mbps	v2.0: 3Mbps
802.11g: 54Mbps	v3.0: 54Mbps
802.11n: 135Mbps	v4.0: 0,3Mbps

Na BLE je možné nahlížet, jako na systém navrhnutý z pohledu nízkých nákladů, tomu odpovídají 3 hlavní body:

1. Pásmo ISM (průmysl, věda a lékařství)

V tomto frekvenčním pásmu je signál pohlcován většinou materiálů, hlavně vodou. Velkou výhodou je, že v tomto pásmu není potřeba získat licence pro provoz. Bohužel to s sebou přináší vysokou saturaci pásma jinými technologiemi (Wi-Fi). Jediná omezení pro operování v tomto spektru jsou výkony vysílačů. I přes své nedostatky je pásmo nejvýhodnější pro provoz Bluetooth LE technologie.



Obrázek 1: Zobrazení pásma a kanálů Bluetooth, spolu s interferencemi Wi-Fi sítí. Přejato z webu <http://www.argenox.com>.

2. Licence IP (licence duševního vlastnictví)

Ta náleží pod skupinu Bluetooth Special Interest Group (Bluetooth SIG). Tato skupina si pro licenční použití klade velmi šedré podmínky, což znamená, že oproti jiným skupinám je cena za licenci velmi malá. S nízkou cenou bluetooth licence klesá i celková cena bluetooth zařízení.

3. Nízká energetická náročnost

Účinnou cestou, jak snížit finální cenu zařízení je ušetřit za materiál, v tomto případě za baterie. Díky nízké energetické náročnosti není potřeba kupovat vysokokapacitní baterie a častá výměna také odpadá. Z toho důvodu byla technologie BLE navržena s ohledem na provoz na knoflíkové baterii.

[1]

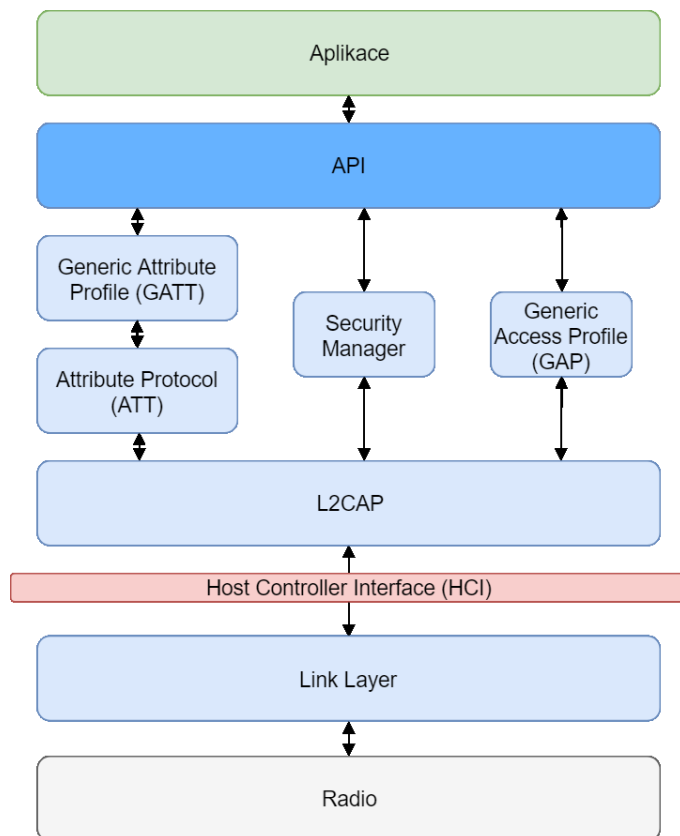
1.1.1 KOMPATIBILITA

Bluetooth LE a Bluetooth jsou dvě odlišné technologie, je proto potřeba zkontrolovat, zda smart zařízení podporuje tuto technologii. U většiny smart zařízení je ovšem brána jako standard. Kompatibilita mezi verzemi je zobrazena v Tabulka 2. Aby mohla komunikace proběhnout, obě zařízení musí podporovat tuto technologii.

Tabulka 2: Kompatibilita mezi standardy bluetooth. [1]

	Single-Mode	Dual-Mode	Classic
Single-Mode	LE	LE	Není možné
Dual-Mode	LE	Classic	Classic
Classic	Není možné	Classic	Classic

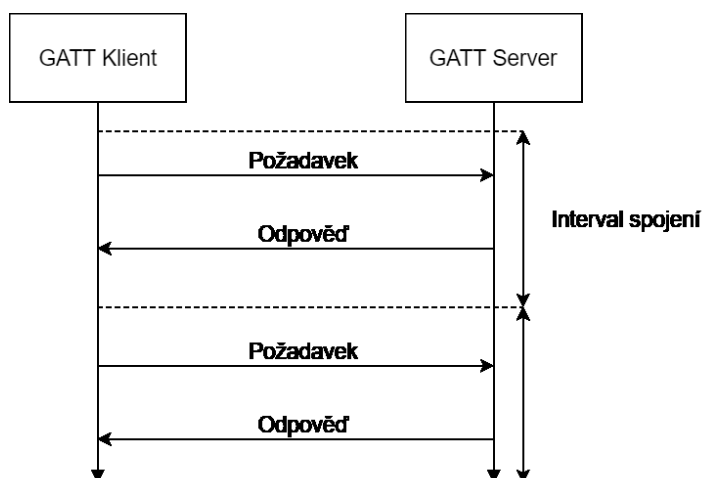
1.1.2 ARCHITEKTURA TECHNOLOGIE



Obrázek 2: Přehled architektury Bluetooth LE.

1.1.2.1 Generic Attribute Protocol (GATT)

GATT je postaven na Attribute Protocol a nastoluje obecný rámec pro data přenášena a ukládána ATT. Dále definuje dvě role: server a klient. Je aktivně použit teprve, až po vytvoření spojení, což znamená, že zařízení již prošlo procesem *advertising*. Připojení ke GATT jsou vždy exkluzivní, znamená to tak, že se serverem může komunikovat maximálně jeden klient.



Obrázek 3: Proces výměny dat mezi serverem a klientem.

Vztah klient/server je pro GATT důležitý, veškeré přenosy jsou započaty klientem. Server mu poté odesílá odpovědi. Dalším pojmem je interval spojení, ten je při připojení navrhnut klientem. Server poté v těchto intervalech kontroluje, zda nepřišly příkazy od klienta.

Datové přenosy pomocí Bluetooth LE jsou založeny na objektech profil, služba a charakteristika. Profil ve skutečnosti neexistuje na serveru samotném, je to pouze předdefinovaná kolekce služeb, sestavená Bluetooth SIG, případně jinými uživateli.

Služby jsou použity k rozdělení dat do logických entit, zaštiťujících data zvaná charakteristiky. Služba může obsahovat více takovýchto charakteristik a je rozlišena pomocí svého specifického UUID. Charakteristika je nejnižší úroveň při přenosu dat, zastřešuje jednu sadu přenášených hodnot. Taktéž má každá charakteristika své specifické UUID.

[1]

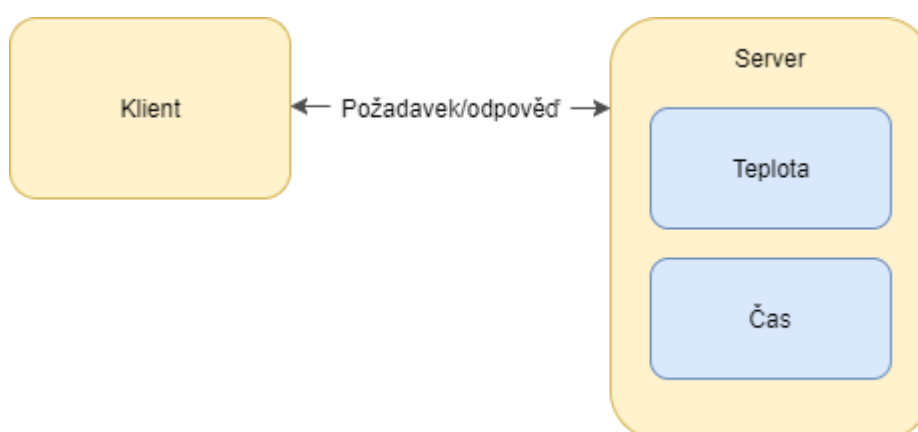
1.1.2.2 Generic Access Profile (GAP)

Generic Access Profile definuje základní společnou funkčnost všech Bluetooth zařízení. Zahrnuje módy a obecné postupy pro objevení blízkých zařízení a připojování se k nim. Gap definuje následující čtyři role:

- *broadcaster*, který pouze odesílá *advertising* pakety,
- *observer* pouze přijímá,
- *peripheral*, je zařízení, které přijímá požadavek na připojení, je přizpůsobeno na pouze jedno aktivní spojení,
- *central* se připojuje, tato role podporuje násobné připojení k více *peripheral*.

1.1.2.3 Attribute Protocol (ATT)

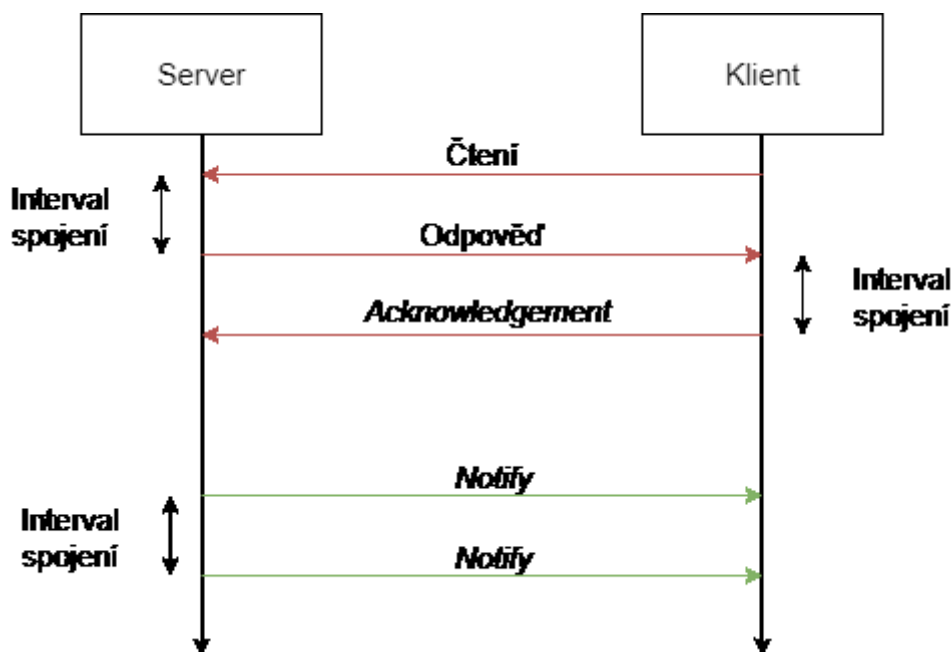
Vytváří možnosti pro přenášení dat. Je závislý na aktivním připojení a umožňuje operace jako *zápis*, *čtení*, *Indicate* a *Notify*. Definuje tak dvě role server a klient.



Obrázek 4: Role zařízení

Klient může přistoupit k hodnotám na serveru pomocí zaslání požadavku, který způsobí odpověď. Pro vyšší účinnost může server odeslat hodnoty bez požadavku pomocí *Notify*, které nevyžaduje odpověď. Další možností je *Indicate*, které již potvrzení vyžaduje.

Acknowledgements (potvrzení) jsou vyžadována po provedení některých operací, to umožňuje zjistit, které pakety byly úspěšně přijaty. Tého metody se využívá, když je potřeba zajistit přijetí paketu. Ovšem jelikož server musí vyčkat na potvrzení, je rychlost přenosu snížena na polovinu.



Obrázek 5: Komunikace s Acknowledgement a bez. Převzato z [1]

1.1.2.4 Security Manager (SM)

Spravuje bezpečnostní otázky při komunikaci Bluetooth. Pomocí definovaných protokolů lze řešit bezpečnost z hlediska párování, generování náhodných kódů pro komunikaci, šifrování dat. A odeslání dat autorizovaných digitálním podpisem během nešifrované či nezabezpečené komunikace.

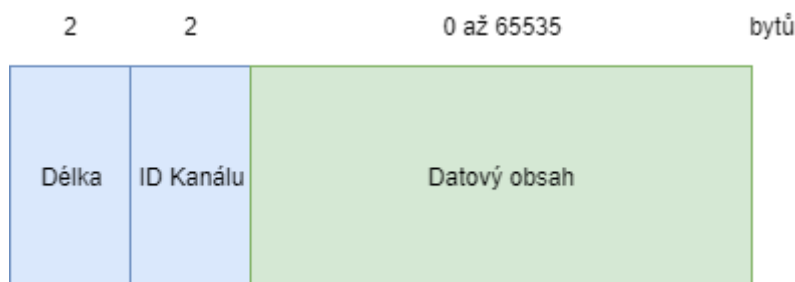
1.1.2.5 Logical Link Control and Adaptation Protocol (L2CAP)

Tato vrstva BLE je využívána k posílání síťových paketů na HCI (Host/Controller Interface) nebo v některých případech přímo na Link Manager.

Tento protokol obstarává:

- přenášení dat pro vyšší vrstvy, může multiplexovat více aplikací skrze jedno spojení;
- rozdělování a spojování paketů;
- může poskytnout jednosměrnou přenosovou správu různých dat pro skupinu ostatních Bluetooth zařízení;

- správu kvality provozu (Quality of Service QoS) pro vyšší vrstvy.



Obrázek 6: Struktura L2CAP paketu. Převzato z [1]

1.1.2.6 Host/Controller Interface (HCI)

Umožňuje hostiteli (Host) zasílat data a příkazy ovladači (Controller) a naopak. Skládá se vlastně ze dvou částí: logického rozhraní a vlastního fyzického rozhraní.

Logické rozhraní definuje příkazy, události a jejich přidružený průběh. Fyzické rozhraní formuluje, jak jsou příkazy, události a data přenášeny pomocí různých možností spojení. Například nadefinovaná spojení obsahují USB (Universal Serial Bus), SDIO (Secure Digital Input/Output) a 2 varianty rozhraní UART (Universal Asynchronous Receiver/Transmitter).

1.2 Výběr Bluetooth LE modulu

Modul byl vybrán s ohledem na jeho budoucí využití (chytrá helma). Prioritně velikost, nízká spotřeba a možnost zpracování analogových a digitálních signálů. Pro tento požadavek byl stanoven standard Bluetooth LE jako vyhovující.

Jako 3 finální moduly byly zvoleny BLE113, BGM113 a BGM123. Hlavní rozdíly uvedeny v tabulce níže.

Tabulka 3: Hlavní rozdíly mezi zvolenými Bluetooth Smart Moduly

	Velikost (mm)	MCU	Počet vstup/výstup pinů	Spotřeba Odeslání(TX)/Přijetí(RX)
BLE113	9.15 x 15.75 x 2.1	CPU architektury 8051	21	TX: 18 mA (0dBm) RX: 14,3 mA
BGM113	9,2 x 15,8 x 1,83	ARM Cortex®-M4	14	TX: 8,8 mA (0dBm) RX: 8,7 mA
BGM123	6.5 x 6.5 x 1.4	ARM Cortex®-M4	30	TX: 8,2 mA (0dBm) RX: 9 mA

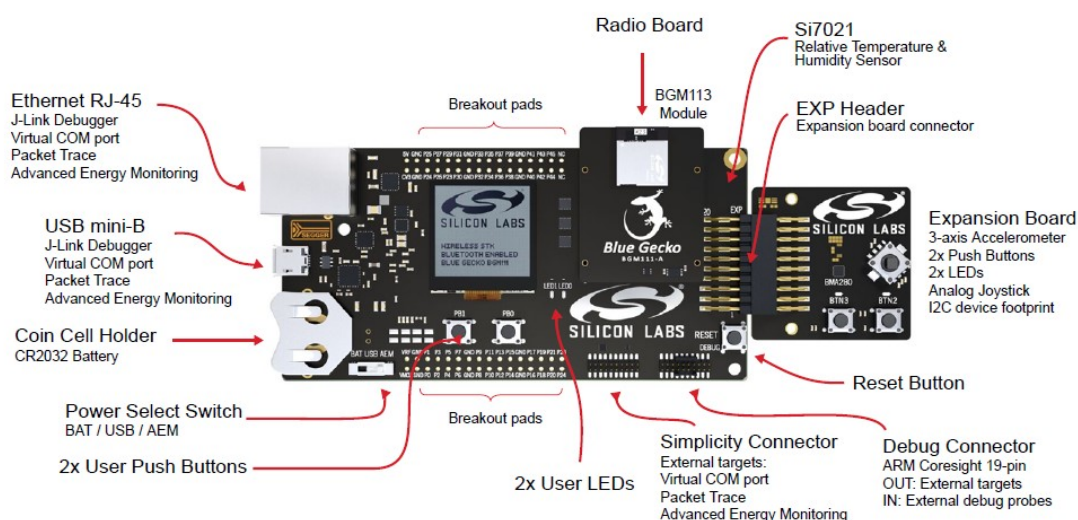
Dalším důležitým prvkem z pohledu bakalářské práce je A/D převodník, jenž je integrován v každém z těchto modulů. Všechny tyto bluetooth moduly obsahují možnost využití až 12—ti bitového převodníku. Maximální vzorkovací frekvence těchto A/D převodníků se pohybuje kolem 10^6 vzorků/s,

což je dostačující. Paměť RAM byla u modulů BGMxxx navýšena na 32 kB, oproti 8 kB modelu BLE113.

Zvolen byl nakonec modul **BGM113**, jakožto modernější a výkonově lepší Bluetooth modul. Velikostně je z pohledu prototypu vhodnější využít modelu většího, protože hardwarově jsou si velmi podobné. Modul proto bude lehčí umístit na desku plošných spojů. Migrace na modul BGM123, pokud bude potřeba velmi malého pouzdra, je možná díky stejnému MCU. Dalším důvodem je z pohledu softwarového vývoje výhoda, že BGM113 a BGM123 využívají stejného vývojářského nástroje, a strukturou firmware jsou stejné. Pro účely budoucího prototypu tak je výhodnější využít BGM113.

1.3 Blue Gecko Module Wireless Starter Kit

Pro usnadnění vytváření softwaru byl pořízen Blue Gecko Module Starter Kit. Tento vývojový kit poskytuje zázemí jak pro vývoj, tak periferní senzory dostupné k testování aplikací. Bluetooth modul se nachází samostatně na Radio Board, lze jej jednoduše připojit k modulu, lze tak testovat více různých zařízení.

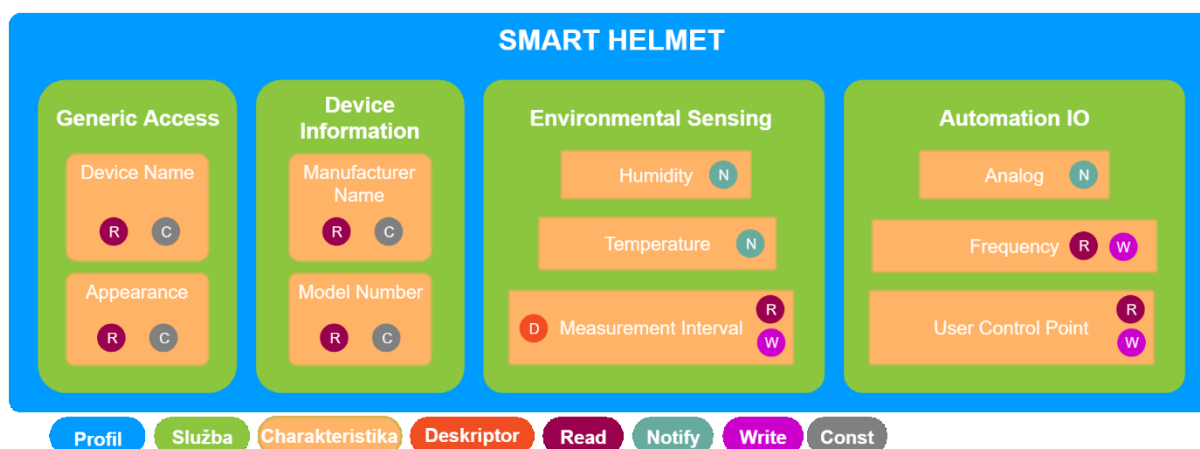


Obrázek 7: Přehled dostupného hardware. Převzato z [2]

Celý modul je napájen a připojen k PC pomocí USB. K debuggování je na starter kitu dostupný J-Link firmy Segger. Jako vývojové prostředí slouží *Simplicity Studio v4*, založené na studiu *Eclipse*.

2 Praktická část

2.1 GATT



Obrázek 8: Struktura navrženého GATT.

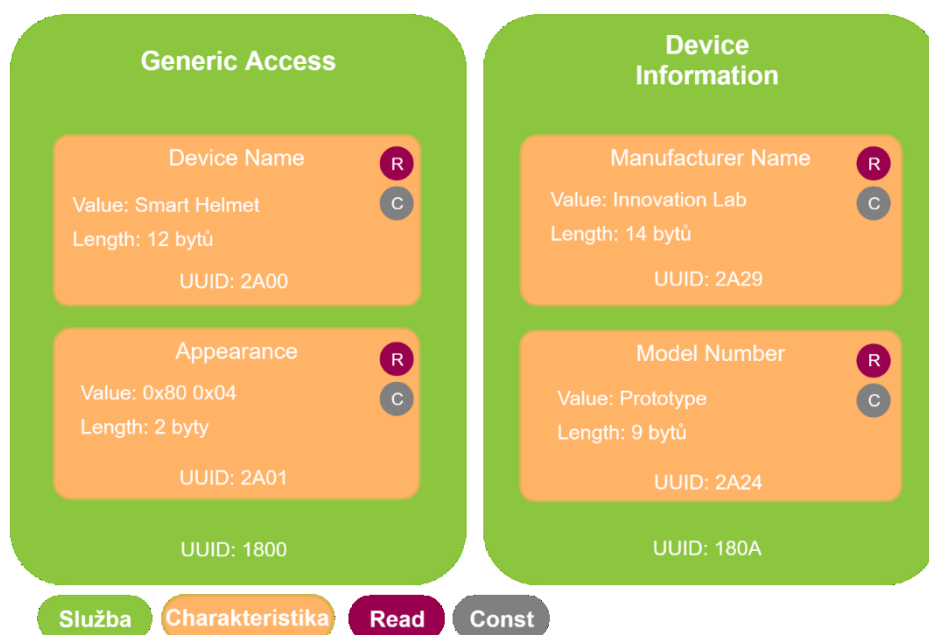
Struktura GATT byla navržena tak, aby bylo možno zasílat výsledky měření z jednotlivých senzorů a A/D převodníku, a aby umožnila:

- nastavení hodnoty intervalu měření pro teplotu a relativní vlhkost,
- frekvenci měření a výběr ze tří vstupních pinů pro analogový signál.

Jako periferní snímač teploty a vlhkosti je použit Si 7021-A20 [3] a tři analogové signály. Bylo dbáno na použití služeb a charakteristik definovaných skupinou Bluetooth SIG [4]. V případě služby Automation IO musely být její charakteristiky upraveny pro potřeby aplikace, neodpovídají proto stanoveným. Výhodou tohoto postupu je fakt, že funkce zařízení, které jsou ovládány a odesílány přes tyto obecné charakteristiky, mohou být použity i smart zařízením, které nevyužívá aplikace, vytvořené v této práci. Lze tak spustit měření teploty a relativní vlhkosti a přijímat výsledky bez použití aplikace vyvinuté pro tuto práci.

Navržený GATT je tvořen jedním profilem, jenž obsahuje veškeré služby. Dvě z nich (Generic Access a Device Information) jsou určeny k identifikaci a kategorizaci zařízení. Následující dvě (Environmental Sensing a Automation IO) se týkají odesílání naměřených výsledků a možnosti nastavení parametrů měření. Kompletní kód GATT je dostupný v Příloha A ve formátu XML. K jeho vytvoření bylo využito grafické rozhraní, jenž je součástí vývojového prostředí *Simplicity Studio v4*. V následujících podkapitolách jsou popsány jednotlivé části navrženého GATT.

2.1.1 GENERIC ACCESS A DEVICE INFORMATION



Obrázek 9: Struktura služeb Generic Access a Device Information.

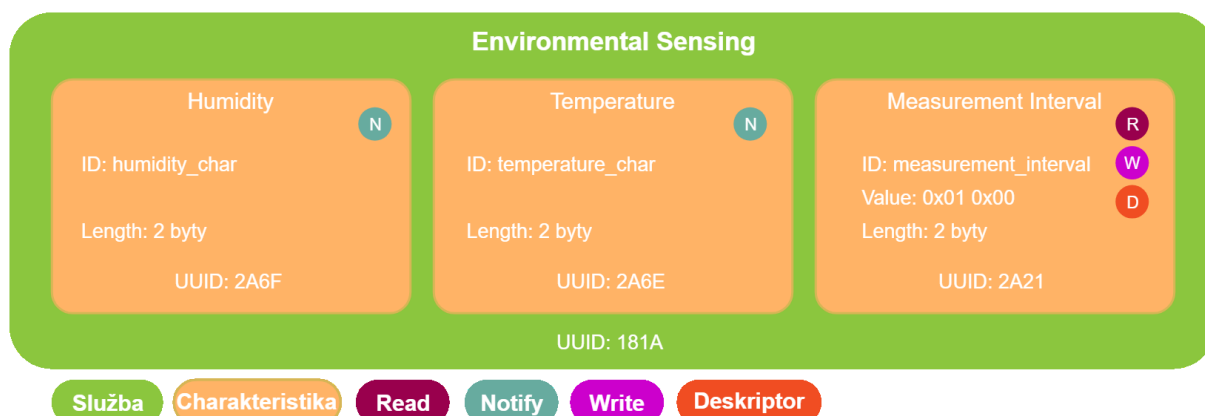
V těchto službách jsou zahrnuty charakteristiky, obsahující základní informace jako jsou Název, jméno výrobce, číslo modelu a externí vzhled, definovaný z knihovny Bluetooth SIG. Charakteristika Device Name je obsažena v paketu pro advertising. Ostatní charakteristiky jsou dostupné na vyžádání pomocí Attribute protocol *Read* k možnosti bližší identifikace.

Každá charakteristika v těchto službách byla použita ze seznamu Bluetooth SIG, formu a druh obsahu tak zobrazí i aplikace jako Bluetooth LE skenery, případně jiné zařízení, které se připojí. Jelikož jsou tyto charakteristiky informační a v průběhu činnosti nenastává potřeba je pozměnit, jsou všechny nastaveny s vlastností *Const* a *Read*. Pro přehlednost byly jednotlivé hodnoty UUID a informace o těchto charakteristikách zobrazeny na Obrázek 9. Jednotlivé charakteristiky v této službě již obsahují své ID, to je využito k odkázání se na danou charakteristiku v kódu aplikace na BGM113.

- **Device Name** je tvořena hodnotou, složenou ze znaků uložených v sekvenci UTF-8, délka této hodnoty je tak 12 bajtů. Tato charakteristika je používána v paketu pro advertising.
- **Appearance** kategorizuje zařízení podle Bluetooth SIG. Předdefinované enumerace zde opět pomáhají možnosti identifikovat zařízení i pomocí Bluetooth LE skenerů nevytvořených speciálně pro účely této aplikace, v případě této práce byla vybrána hodnota odpovídající *Cycling*.
- **Manufacturer Name**, tato charakteristika obsahuje jméno výrobce, pomáhá blíže specifikovat zařízení. Podobně jako Device Name je uloženo ve formě UTF-8.
- **Model Number** byla zahrnuta pro budoucí možnost rozlišení verzí.

Tyto služby jsou primárně využity k umožnění identifikace externími smart zařízeními, případně jinými přístroji, komunikujícími pomocí Bluetooth LE. Částečně se promítají do paketu pro advertising. Přístup k informacím je umožněn pomocí vlastnosti *Read*.

2.1.2 ENVIRONMENTAL SENSING



Obrázek 10: Struktura služby Environmental Sensing.

Environmental Sensing definuje, jak jsou odesílány výsledky z digitálního senzoru a jak je možno nastavit samotný interval měření ze smart aplikace. Pro zajištění komunikace se zařízením bylo potřeba využít tři charakteristik definovaných Bluetooth SIG. Další možností bylo využít služby Health Thermometer. Ovšem obsažené charakteristiky byly oproti zvoleným pro účely odesílání dat z digitálního senzoru Si 7021-A20 zbytečně obsáhlé.

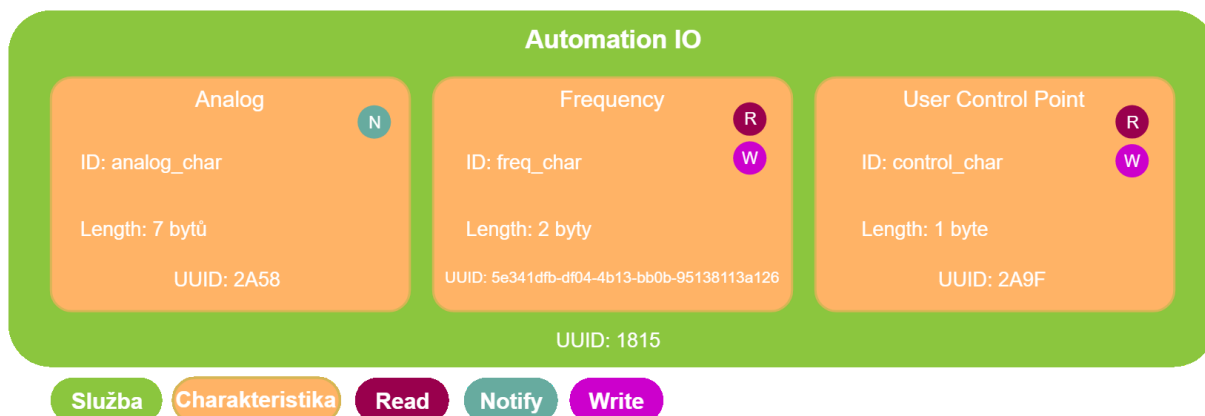
Pomocí Si 7021-A20 získáme teplotu a relativní vlhkost. Pro tyto naměřené hodnoty byly ze seznamu definovaných charakteristik vybrány tyto:

- **Humidity**, tato charakteristika obsahuje hodnotu naměřenou ve formátu 16—ti bitového neznaménkového integeru, očekávanou jednotkou jsou procenta. Hodnota je uložena v mocnině formě 10^{-2} , tato přesnost je vyžadována, jelikož tak byla definována Bluetooth SIG. Přesnost senzoru Si 7021 je ovšem na rozsahu 0 % — 80 % je ± 3 % podle katalogového listu [3]. Jako operace z Attribute protocol byla zvolena *Notify*.
- **Temperature** charakteristika je podobná Humidity, až na 16—ti bitový znaménkový integer. Ten je využit, jelikož uložená hodnota používá jednotku $^{\circ}\text{C}$. Hodnota je vepsána ve formě 10^{-2} . K odeslání hodnoty slouží opět *Notify*.

Dále byla vybrána charakteristika, pomocí které je externí aplikace schopna nastavit parametry měření, respektive interval mezi měřeními:

- **Measurement Interval** obsahuje hodnotu ve formátu 16—ti bitového neznaménkového integeru, tato hodnota je v jednotce sekunda. Pro přístup externí aplikace k aktuální hodnotě měřicího intervalu je nastavena vlastnost *Read* na true. Nová hodnota je do charakteristiky zapsána pomocí *Write*.

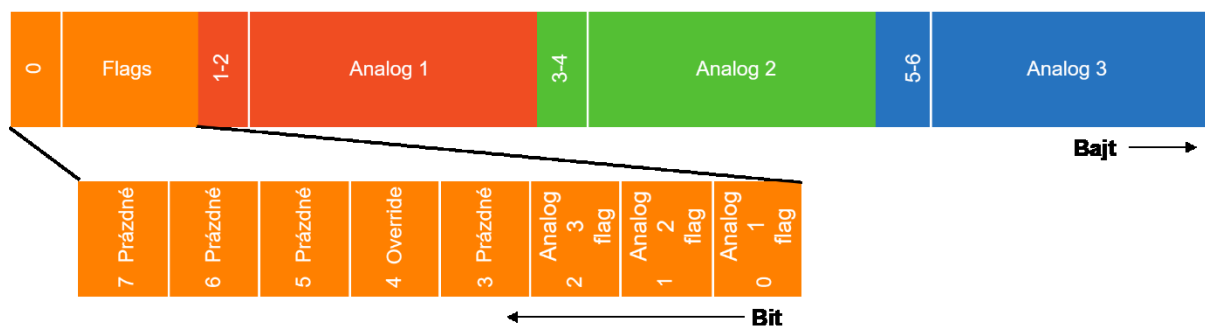
2.1.3 AUTOMATION IO



Obrázek 11: Struktura služby Automation IO.

Tato služba obsahuje charakteristiky pro přenos naměřených hodnot z A/D převodníku, byla navržena, aby bylo možné měřit a přenášet až tři různé analogové signály najednou. Pro odesílání tří různých signálů byla upravena charakteristika Analog, pro nastavení frekvence A/D převodníku byla přidána Frequency a User Control Point slouží pro komunikaci zvoleného pinu a módu měření. V následujících podkapitolách jsou blíže rozvedeny jednotlivé charakteristiky.

2.1.3.1 Struktura charakteristiky Analog



Obrázek 12: Struktura upravené charakteristiky Analog.

Návrh struktury musel umožnit přenos výsledků A/D převodníku v rozumné rychlosti. Signál by neměl být podvzorkován, proto byly zjištěny minimální použitelné vzorkovací frekvence pro jednotlivé vstupní signály. Pro EKG, za účelem získání frekvence srdečního rytmu, je minimální vzorkovací frekvence 50 Hz [5]. Pro zjištění stejné informace ze signálu fotopletysmografie je podle [6] dostačujících 25 Hz.

Pro tři signály nelze vytvořit samostatné charakteristiky z důvodu technických limitací technologie Bluetooth LE, která nebyla navržena pro přenos většího objemu dat za krátký interval [1]. Mezi limitace ovlivňující rychlost přenosu patří:

- interval spojení,
- délka paketu,
- velikost maximální přenosové jednotky (MTU),

- možnost přijmutí více paketů za jeden interval spojení,
- použitá operace z Attribute protocol.

[8]

Zjištěné specifikace testovaného spojení (mezi Bluetooth LE modulem BGM113 a tabletem Google Nexus 7, Android 6.0.1) při použití navrženého GATT:

Tabulka 4: Specifikace spojení BGM113 a tabletu Google Nexus 7.

Interval spojení	39 ms (12 ms)
Paketů za interval	6
Maximální přenosová jednotka	247 bajtů

Specifikace interval spojení a maximální přenosová jednotka byly zjištěny na zařízení BGM113, jehož softwarová knihovna (*Gecko SDK Suite v2.1.1: Bluetooth 2.8.0.0*) obsahuje událost, zobrazující tyto hodnoty při iniciaci spojení (*evt_le_connection_parameters*) [7]. Minimální interval spojení pro Google Nexus 7 není uveden v oficiálních specifikacích Android ani v informacích výrobce tabletu, podle [9] se ovšem liší dle použitého zařízení. Nejnižších hodnot dosahují zařízení s operačním systémem Android a to 7,5 ms, což je i nejnižší možná hodnota pro Bluetooth LE spojení [10].

Hodnota v závorce z Tabulka 4 představuje interval spojení po odeslání požadavku na jeho snížení ze systému Android (metoda *requestConnectionPriority()* třídy *BluetoothGatt* viz 2.3.3.4. Snížení intervalu spojení ovšem mělo na testovací soustavě za následek značnou nestabilitu. Ke ztrátě spojení docházelo v intervalech pěti vteřin od jeho navázání. Nižší interval proto jako možnost urychlení přenosu, lze využít, ovšem stabilita spojení se snižuje.

Dalším intervalem ovlivňujícím rychlost přenosu je *Acknowledgement* viz 1.1.2.3, z tohoto důvodu je zvolenou operací k odeslání dat zvoleno *Notify*.

Počet maximálního možného počtu přijatých paketů za jeden interval spojení byl zjištěn odesláním paketů obsahujících pořadové číslo daného odeslání a následným sledováním ztracených paketů v jednom intervalu. Za jeden interval spojení tak dokázal tablet přijmout 6 paketů.

Při použití tří samostatných charakteristik, s možností různých vzorkovacích frekvencí, by docházelo ke ztrátě paketů a tím i dat z A/D převodníku:

$$\text{Maximální propustnost} = \frac{1\,000\,ms \times \text{vzorků v paketu} \times \text{pakety za interval}}{\text{interval spojení}} \quad (1)[8]$$

$$\text{Maximální propustnost} = \frac{1\,000 \times 1 \times 6}{39} = 153 \text{ vzorků za vteřinu} \quad (2)$$

V případě použití samostatné charakteristiky pro každý vstupní signál A/D převodníku, by docházelo k překročení hodnoty z rovnice (2) už při vzorkování tří signálů frekvencí 50 Hz, což je vzhledem k požadavkům převáděných signálů nepřijatelné.

Sloučením všech naměřených hodnot do jednoho paketu bylo umožněno využít maximální frekvence z rovnice (2), pro vzorkování všech tří signálů najednou. Ukládání do paketu je řešeno dle

přiřazených bajtů viz Obrázek 12 ve formátu little-endian. Jako důsledek tohoto návrhu tak nelze pro různé signály v průběhu jednoho měření použít různé vzorkovací frekvence.

Návrh ovšem v případě měření jediného analogového signálu umožňuje využití jinak prázdných bajtů, rezervovaných pro výsledky dalších vstupů. Tato metoda tak umožňuje v případě potřeby navýšit maximální možnou vzorkovací frekvenci jednoho analogového signálu až na trojnásobek. A to navýšením hodnoty vzorků v paketu z jedné na tři, viz rovnice (1), tato metoda byla pro přehlednost pojmenována *Override*.

Prvním bajtem v Analog je *Flags*, ten slouží, jak je zobrazeno na Obrázek 12, k předání informace o obsahu odeslaného paketu přijímajícímu zařízení. První tři bity jsou rezervovány pro analogové signály, kdy hodnota bitu 1 znamená, že signál je obsažen, naopak hodnota 0 vyjadřuje absenci hodnoty v paketu. Bit 4 je rezervován pro informaci, zda je povolen mód *Override* viz 2.2.5.2.

2.1.3.2 Frequency

Charakteristika *Frequency* je použita pro nastavení vzorkovací frekvence A/D převodníku. Databáze Bluetooth SIG neobsahuje charakteristiku požadované funkce, pouze již využitou *Measurement Interval*, která je definována jako interval v sekundách. Duplicitní hodnota je povolena, ovšem pro účely práce bylo praktičtější navrhnout novou unikátní charakteristiku. Ta má své nové UUID, které již nepatří do předdefinovaných, proto je na Obrázek 12 uvedena v celé délce.

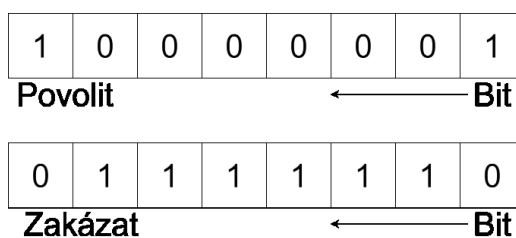
Pro přístup a možnost zapsání jsou povoleny *Read* a *Write*. Hodnota je tvořena 16—ti bitovým neznaménkovým integerem s jednotkou Hertz.

2.1.3.3 User Control Point



Obrázek 14: Struktura charakteristiky User Control Point.

Charakteristika *User Control Point* navržena jako ovládací přístup k volbě vstupního pinu A/D převodníku a povolení režimu *Override*. Je založena na využití bitových operací, příklad bajtu pro volbu činnosti A/D převodu signálu analog 1:



Obrázek 13: Příklad bajtů pro volbu pinu pro A/D převodník.

Operace povolit předá bit k provedení bitového součtu, naopak operace zakázat očekává provedení bitového součinu viz 2.2.3.3.

2.2 Firmware BGM113

Pro vývoj firmware bylo rozhodováno mezi:

1. Skriptovací jazyk *BGScript*, vytvořený firmou Silicon Labs, speciálně pro tyto účely.
2. Programování firmware v jazyku C/C++ ve vývojovém prostředí *Simplicity Studio v4*, vyvinutém toutéž firmou.

Vybráno bylo prostředí *Simplicity Studio v4*, protože v průběhu zadání práce byla firmou Silicon Labs oficiálně ukončena podpora BGScript na všech produktech založených na mikrokontroleru rodiny EFR32. A to v knihovnách Bluetooth Smart Software verze 2.4.0 a výše [11]. Jeho použití by tedy znamenalo vytvoření zastaralého, již nepodporovaného řešení, bez možnosti budoucí návaznosti na dosažené výsledky. Dále programování v C/C++ umožňuje hlubší přizpůsobení některých použitých ovladačů a obecně větší volnost pro přístup k řešení. K sepsání kódu byly využity nejnovější dostupné vývojové nástroje Gecko SDK Suite v2.1.1: Bluetooth 2.8.0.0.

Pro přístup ze smart zařízení byla pro účely testování použita aplikace z Google Play, *nRF Connect for Mobile*, vyvinutou společností Nordic Semiconductor.

Následující podkapitoly jsou primárně řazeny dle vnitřní struktury firmware a jeho zdrojových souborů. Zdrojový kód napsaného firmware je dostupný z Příloha A.

2.2.1 STRUKTURA PROJEKTU

Celý firmware je založen na prázdném příkladu soc-empty, vytvořeným Silicon Labs. Ten obsahuje minimální projektovou strukturu potřebnou k vytvoření funkční Bluetooth LE aplikace v jazyce C. Knihovna pro GATT musí být vytvořena separátně a poté připojena. Projekt obsahuje základní direktiva pro preprocesor:

- základní knihovny pro iniciaci a funkci zařízení,
- dále knihovny pro funkci zásobníku Bluetooth LE,
- následují knihovny obsahující výchozí hodnoty pro nastavení zásobníku gecko,
- jako další jsou knihovny, které provádí inicializaci. Pro účely práce musela být tato knihovna doplněna o povolení pro potřebné periferie a součásti Bluetooth modulu,
- poté definuje makro pro maximální počet souvislých spojení. Výchozí hodnota jsou čtyři spojení, pro účely projektu byla změněna na jedno. Makro totiž v další části projektu ovlivňuje, kolik vnitřní paměti bude alokováno pro potřeby Bluetooth zásobníku a více než jednoho souvislého spojení není pro účel této práce potřeba. To zároveň zabraňuje nechtěným interakcím a změnám globálních proměnných v průběhu měření.

K této části byly připojeny další direktiva, jenž jsou využita v práci:

- ovladače pro digitální senzor teploty a relativní vlhkosti Si 7021-A20,
- makra pro typové převody,
- knihovna vytvořená pro práci s A/D převodníkem,
- enumerace a definice pro práci s časovači.

Tato část je následována konfigurací Bluetooth zásobníku. Nejprve je vypočítán jemu přidělený potřebný prostor. Hodnota je poté předána dále. Oproti výchozímu nastavení byla zakázána vlastnost *sleep.flags*, jenž umožňuje vstup do módu hlubokého spánku (součást blokovací události popsané v 2.2.2). Povolení tohoto režimu umožní pozastavení vysokofrekvenčních oscilátorů modulu, využívaných A/D převodníkem. To znamená nemožnost dokončit převod hodnoty a aplikace zůstane zacyklena viz kapitola 2.2.5. Další vlastnosti byly ponechány na výchozí hodnotě, pro potřeby práce je není třeba měnit.

Následuje main. Nejprve potřeba volat tři funkce, nastavující Bluetooth modul. Při tomto procesu musí být dodrženo specifické pořadí [12]:

- 1) *initMcu()* zapne oscilátory a nakonfiguruje energetické módy,
- 2) *initBoard()* je použita k inicializaci součástí modulu, jako externí flash paměti,
- 3) *initApp()* pro tuto aplikaci povolí a nakonfiguruje piny pro i²c.

Bez nich nelze zařízení používat. Před vstupem do hlavního cyklu musí být volána funkce, která aplikuje sestavenou konfiguraci Bluetooth zásobníku. Poté jsou volány funkce týkající se digitálního senzoru a A/D převodníku.

Následuje nekonečný cyklus, který spravuje veškerou funkčnost aplikace. Celá vnitřní architektura je zaměřena na zpracování událostí v této hlavní smyčce. Události jsou přijímány asynchronně. Na začátku cyklu je potřeba vytvořit pointer na událost, kterému ji následně přiřadíme pomocí Event Listener. Pomocí *handleru* je sestavena celá funkčnost aplikace.

2.2.2 VOLBA EVENT LISTENER

Event Listener musí být implementován po vytvoření pointeru, kterému přiřazuje událost. Pro účely projektu byla využita funkce *gecko_wait_event()*. Ta blokuje cyklus, dokud nepříjde nová událost do řady. Během „čekání“ na příchozí událost implementuje energetické módy, čím snižuje spotřebu. Hlavní výhodou tohoto řešení je, že spotřeba je snižována automaticky, bez potřeby jakékoliv další implementace ze strany návrhu. Nevýhodou naopak je, že pokud je povolen *deep_sleep*, nelze využít A/D převodníku.

Další možností je neblokující Event Listener. Na rozdíl od předchozí funkce nečeká na příchozí událost a vrací hodnotu NULL. Ovšem zde by musely být energetické módy zvládány manuálně v aplikaci. Pro potřeby práce tak byla *gecko_wait_event()* vyhovující.

2.2.3 TŘÍDĚNÍ PŘÍCHOZÍCH UDÁLOSTÍ A JEJICH STRUKTURA

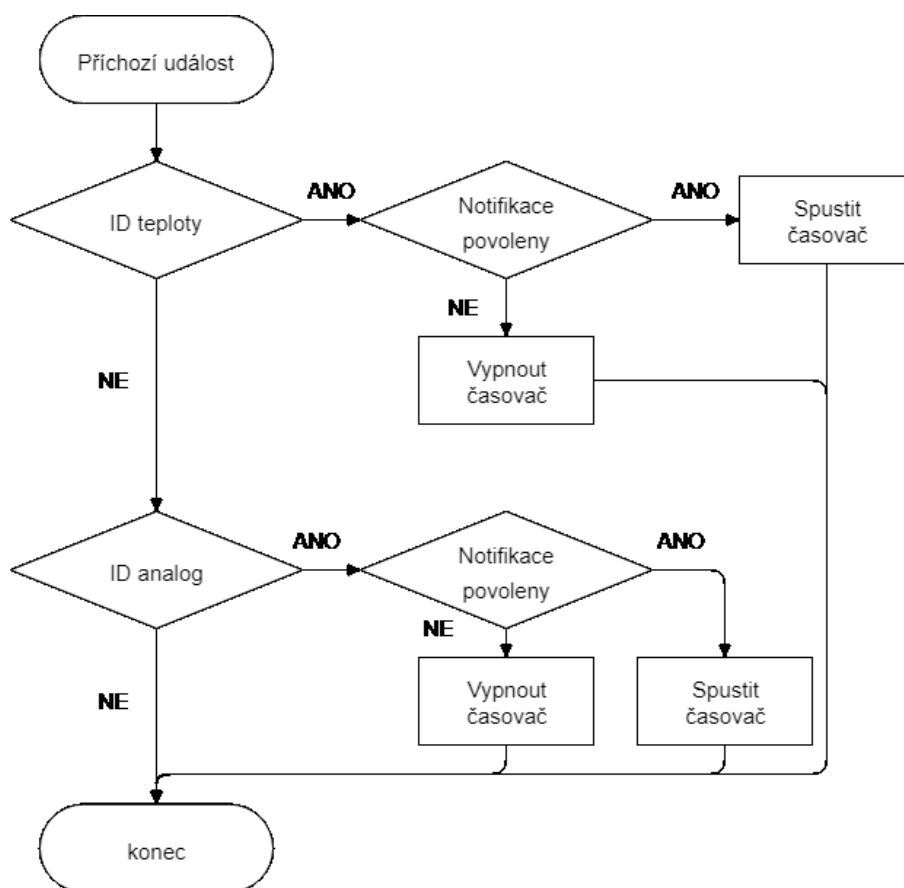
Tato část se věnuje událostem, pomocí kterých je sestavena celá finální funkčnost firmware. Rozebírá strukturu a implementaci. Události jsou tříděny pomocí příkazu *switch* podle hlavičky. Následující podkapitoly znázorňují jednotlivé *case*.

2.2.3.1 *gecko_evt_system_boot_id*

Při zapnutí aplikace je *gecko_evt_system_boot_id* jednou z prvních událostí, která přijde do řady. Indikuje úspěšný start a připravenost Bluetooth LE rádia. V této části je nastaven a započat *advertising*. Aplikace se tak stane viditelnou hned po zapnutí. Interval odesílání paketů *advertisingu* byl nastaven na 100 ms. Doba, po kterou je vysíláno bylo nastaveno na neurčito. Další konfigurace odpovídají obecným protokolům pro nalezení a připojování.

2.2.3.2 *gecko_evt_gatt_server_characteristic_status_id*

Povolení operace *Notify* Attribute Protokolu u dané charakteristiky vyvolá událost *gecko_evt_gatt_server_characteristic_status_id*. Struktura příchozí události obsahuje potřebné informace pro rozřídění. Prvními třídícími parametry jsou ID charakteristiky a zda u ní došlo ke změně. Poté je zjištěno, zda bylo povoleno nebo zakázáno *Notify*.



Obrázek 15: Ovládání časovačů měření.

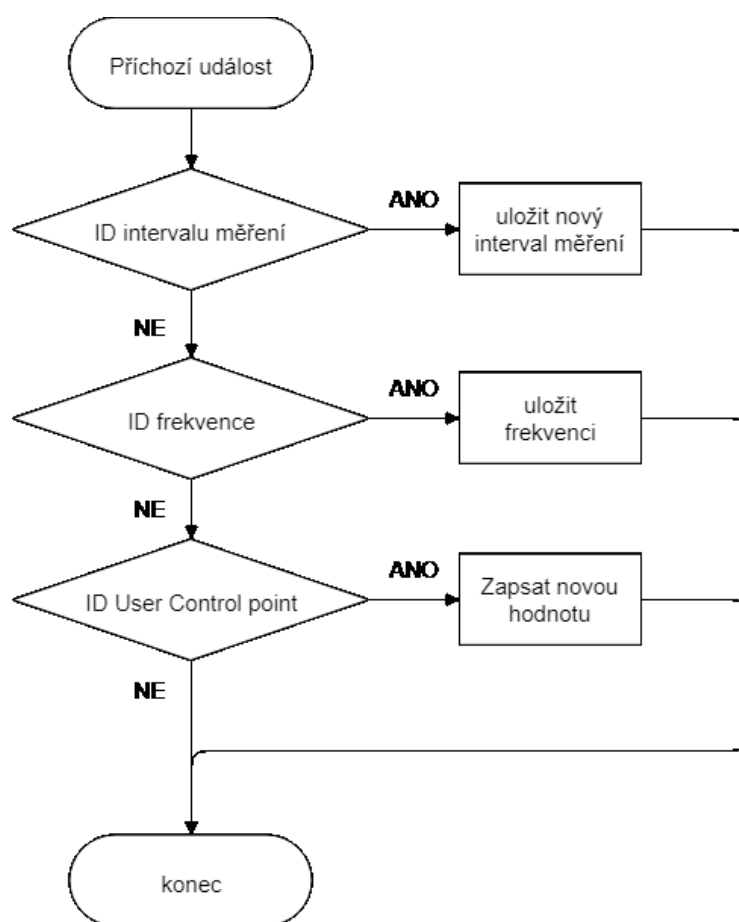
ID teploty a analog z Obrázek 15 značí podmínky *if* a *else if*. Výsledek pravda znamená, že se jedná o danou charakteristiku a zároveň, že došlo k změně nastavení *Notify*. Následně je zkontrolováno, zda došlo k jejímu povolení či zakázání.

Změna pro charakteristiku zasílající data o relativní vlhkosti není zachycena. Tato volba byla učiněna, protože digitální senzor měřící teplotu a vlhkost, měří při jednom měření jak teplotu, tak vlhkost viz 2.2.4. Bylo tak rozhodnuto pro zpracování a odeslání obou výsledků. To znamená jeden společný

interval. Ze stejného důvodu proto není umožněno měřit a odesílat pouze výsledky o měření relativní vlhkosti.

Jednotlivé časovače jsou instancí `cmd_hardware_set_soft_timer`. Nastaveny jsou podle globálních proměnných vytvořených pro tento účel viz podkapitola 2.2.3.3. V pravidelných intervalech vyvolávají události, které jsou opět třízeny v této smyčce. Identifikace příchozí události je možná díky hodnotě `handle`, která nese číselné označení daného časovače. Enumerace pro tyto hodnoty lze nalézt v hlavičkovém souboru `soft_timer_def.h`. Jelikož dochází ke spuštění časovačů v této události, nelze v průběhu aktivního měření změnit frekvenci případně interval mezi vzorky. Je to tak potřeba provést před touto událostí. V případě vypnutí notifikací pro `analog` je resetována proměnná týkající se režimu *Override* viz kapitola 2.2.5.2.

2.2.3.3 `gecko_evt_gatt_server_attribute_value_id`



Obrázek 16: Switch pro třídění události změny hodnoty.

Pro kontrolu změny hodnoty v lokálním GATT je možné použít dvě metody. První metodou je pravidelné čtení GATT pomocí vlastního časovače. Zde by ovšem nastával problém, jelikož návratová struktura při vyžádání hodnoty nevrací ID charakteristiky. Musel by tak být implementován další časovač s vlastností single-shot (spustí se pouze jednou), který by identifikoval příchozí odpověď z GATT.

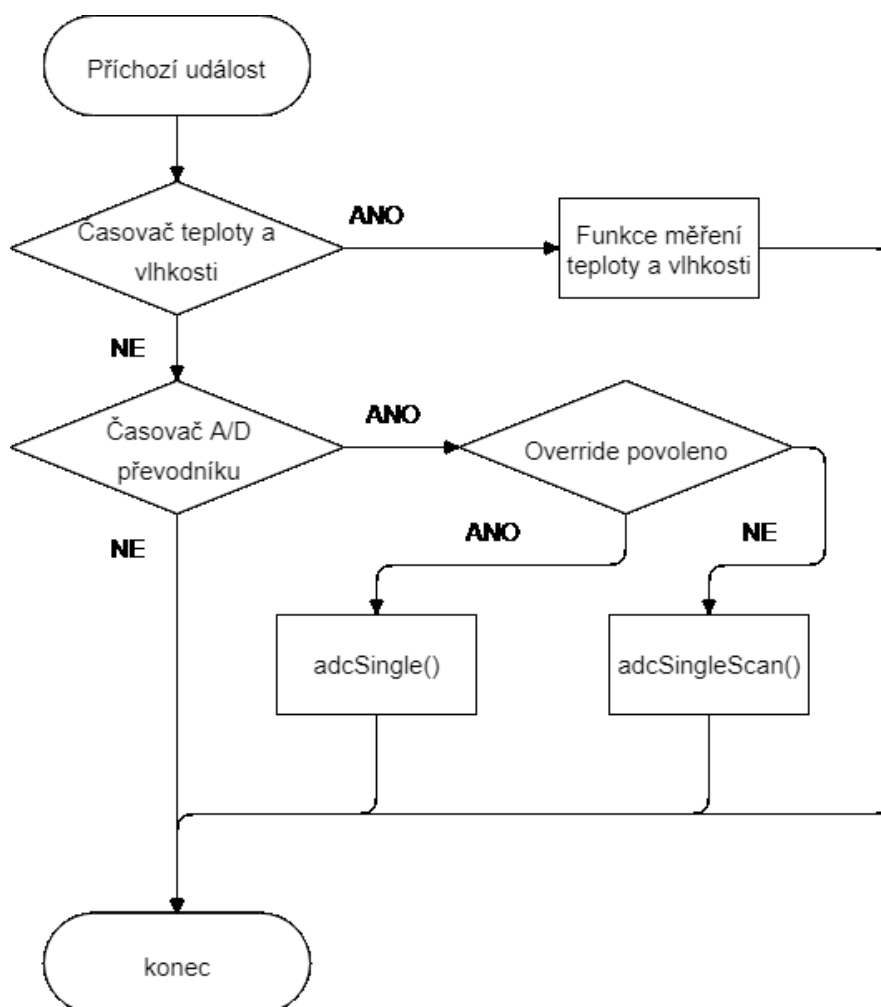
Další a mnohem jednodušší možností je využít události indikující změnu hodnoty. Tak získáme informaci o změněné hodnotě i charakteristice.

V této události jsou tedy viz Obrázek 16 spravovány části, týkající se nastavení parametrů měření. Frekvence a interval měření jsou ukládány do globální proměnné. Následně jsou použity v případě popsaném v podkapitole 2.2.3.2.

Část týkající se User Control Point umožňuje přístup k hodnotě, definující volbu pinů a dalšího nastavení A/D převodníku. Struktura příchozí hodnoty je popsána v podkapitole 2.1.3.1. Prvně se tak zkontroluje stav nejvýznamnějšího bitu (MSB), zda jde o povolení či zakázání. V případě povolení je proveden bitový součet s globální proměnnou *control_point*, výsledek je ještě upraven o vynulování MSB. Zakázání provede bitový součin.

2.2.3.4 *gecko_evt_hardware_soft_timer_id*

Události vyvolané časovači ovládanými v předchozí části, jsou zde zpracovány. K identifikaci je ve struktuře události dostupný *handle*, viz kapitola 2.2.3.2.



Obrázek 17: Struktura case pro časovače.

Jednotlivé funkce z Obrázek 17 jsou zpracovány v kapitole 2.2.5. Pro měření teploty a vlhkosti je návrh přímočarý. U události časovače A/D převodníku je potřeba zkontrolovat nastavený typ převodu. Je tedy provedeno zjištění hodnoty bitu, nesoucího informaci, o režimu *Override* viz 2.2.5.2. Pak je použita příslušná funkce A/D převodníku.

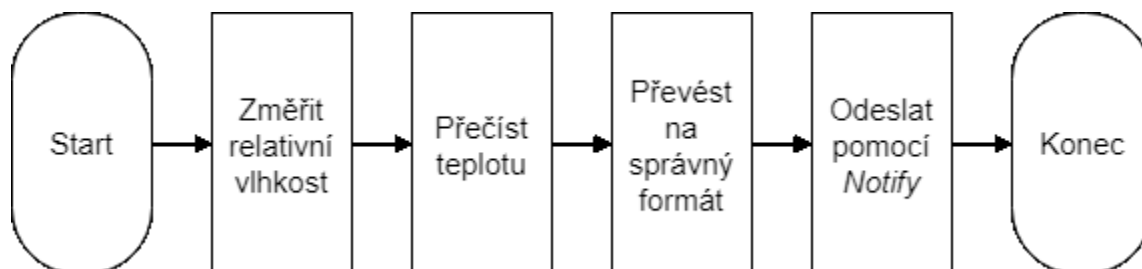
2.2.3.5 *gecko_evt_le_connection_closed_id*

V případě ztráty spojení, je potřeba zastavit časovače a resetovat globální proměnné na výchozí hodnoty. K tomuto úkolu je využita událost *gecko_evt_le_connection_closed_id*. Dále je potřeba restartovat advertising, s hodnotami nastavenými v události při zapnutí zařízení. Po provedení této rutiny je BGM113 připravené k novému připojení

2.2.4 FUNKCE PRO VYUŽITÍ DIGITÁLNÍHO SENZORU SI 7021

Pomocí *temp_humid_Measure()* jsou naměřena a odeslána data z digitálního senzoru. Využívá ovladačů, které jsou součástí *Blue Gecko Development Kit*. Základem je komunikace s Si 7021 pomocí sběrnice I²C. Ta je inicializována v *initApp()*, pomocí knihovny *i2cpm.h*. Zde jsou obsaženy definice pro nastavení datových kanálů pro data (SDA) a časovač (SCL). Výchozí hodnoty této knihovny musí být přepsány za nové, specifické pro každý Bluetooth LE modul. Ty byly obsaženy v nové knihovně *hal-config-board.h*, která byla vytvořena pomocí nástroje *Hardware Configurator* (součást vývojového rozhraní *Simplicity Studio v4*).

Dle datasheetu BGM113 je pro linku SDA využit pin PC10 a pro SCL PC11. K těmto pinům je také připojen Si 7021. Po konfiguraci I²C je možné využít softwarových ovladačů k tomuto senzoru.



Obrázek 18: Vývojový diagram funkce pro měření teploty a vlhkosti.

K samotnému měření a získání výsledků je použita *MeasureRHAndTemp()*. Té jsou předány pointery na bajtová pole pro uložení výsledků, adresa a jaké rozhraní I²C má být použito, lze totiž vytvořit několik nastavení rozhraní. Funkce tak sestaví komunikační bajty. Nejprve odešle příkaz pro naměření nové hodnoty a následně přečte naměřenou hodnotu. Jako první je měřena relativní vlhkost, jelikož k jejímu naměření musí senzor naměřit i teplotu. Tu následně také přečte. V případě, že jsou data úspěšně přečtena, případně dojde k přesažení hodnoty maximálního povoleného času pro čtení, dojde k návratu do těla funkce *temp_humid_Measure()*. I když Si 7021 vrací hodnoty ve formě 16—ti bitových neznaménkových integerů, jsou z důvodů nutnosti použít převodové funkce uloženy do 32—bitových integerů. Proto se zbylá část zabývá typovým převodem na formát definovaný GATT viz 2.1.2.

Po úspěšném získání hodnot ve správném formátu jsou hodnoty odeslány pomocí *gecko_cmd_gatt_server_send_characteristic_notification*.

2.2.5 FUNKCE PRO A/D PŘEVODNÍK

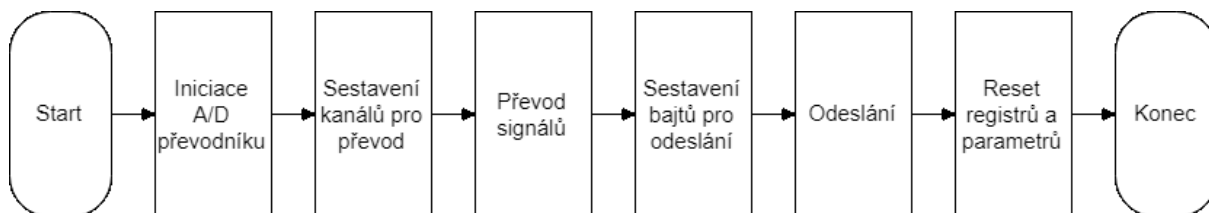
Při návrhu funkcí pro A/D převodník, bylo dbáno na maximální využitelnost navržené charakteristiky pro komunikaci se smart zařízením viz 2.1.3.1. Z tohoto důvodu byly implementovány a upraveny dvě funkce, nalezené v [13]. Tyto a další jsou součástí vytvořené knihovny *adc.h*, dostupné v příloze Příloha A.

Aby bylo možné využít A/D převodník, je potřeba spustit jeho časovač a inicializovat hodnoty porovnávací hladiny pro použité typy převodů. To zajišťuje *adcSetup()*, která je volána před vstupem do hlavního cyklu. A na konci, po převedení a odeslání hodnoty, je potřeba resetovat registry a parametry vztahující se k převodu pomocí *adcReset()*.

Navržené řešení umožňuje měřit libovolný analogový signál, splňující Shannonův teorém. Pro výchozí režim je maximální vzorkovací frekvence 150 Hz viz 2.1.3.1. Pro možnost souběžného měření teploty a relativní vlhkosti digitálním senzorem, aniž by docházelo ke ztrátě paketů je potřeba nastavit maximálně 100 Hz. V režimu *Override* lze využít pro vzorkování až 450 Hz, opět v případě využívání pouze jediného aktivního měření. Vstupní signál musí být v rozpětí 0 V – 5 V, jiné hodnoty by mohly mít za následek poškození modulu. Nelze měřit záporné napětí. V případě povoleného režimu *deep_sleep* v nastavení zásobníku Bluetooth, selže převod, při pokusu inicializace časovačů.

2.2.5.1 Výchozí funkce použitá pro A/D převodník

Jako výchozí metoda pro převod analogové hodnoty, je používána *adcSingleScan()*. Ta vždy proměří napětí na všech zvolených pinech, pro tuto práci tedy PF0, PF1 a PD14.



Obrázek 19: Funkce *adcSingleScan()*.

Při činnosti této funkce jsou vždy převedeny hodnoty ze všech vstupů, i když k nim není připojen žádný signál. To znamená, že všechny bajty nesoucí hodnotu mohou být plné, i když nebyly vyžádány externí aplikací, v tomto případě hraje roli nultý bajt charakteristiky. Ten obsahuje informaci o obsahu hodnot.

Struktura je popsána na Obrázek 19. Inicace znamená nastavení minimálního času potřebného k přivedení převodníku do povoleného stavu. Poté je nastavena reference a jsou sestaveny kanály podle [13] a [14].

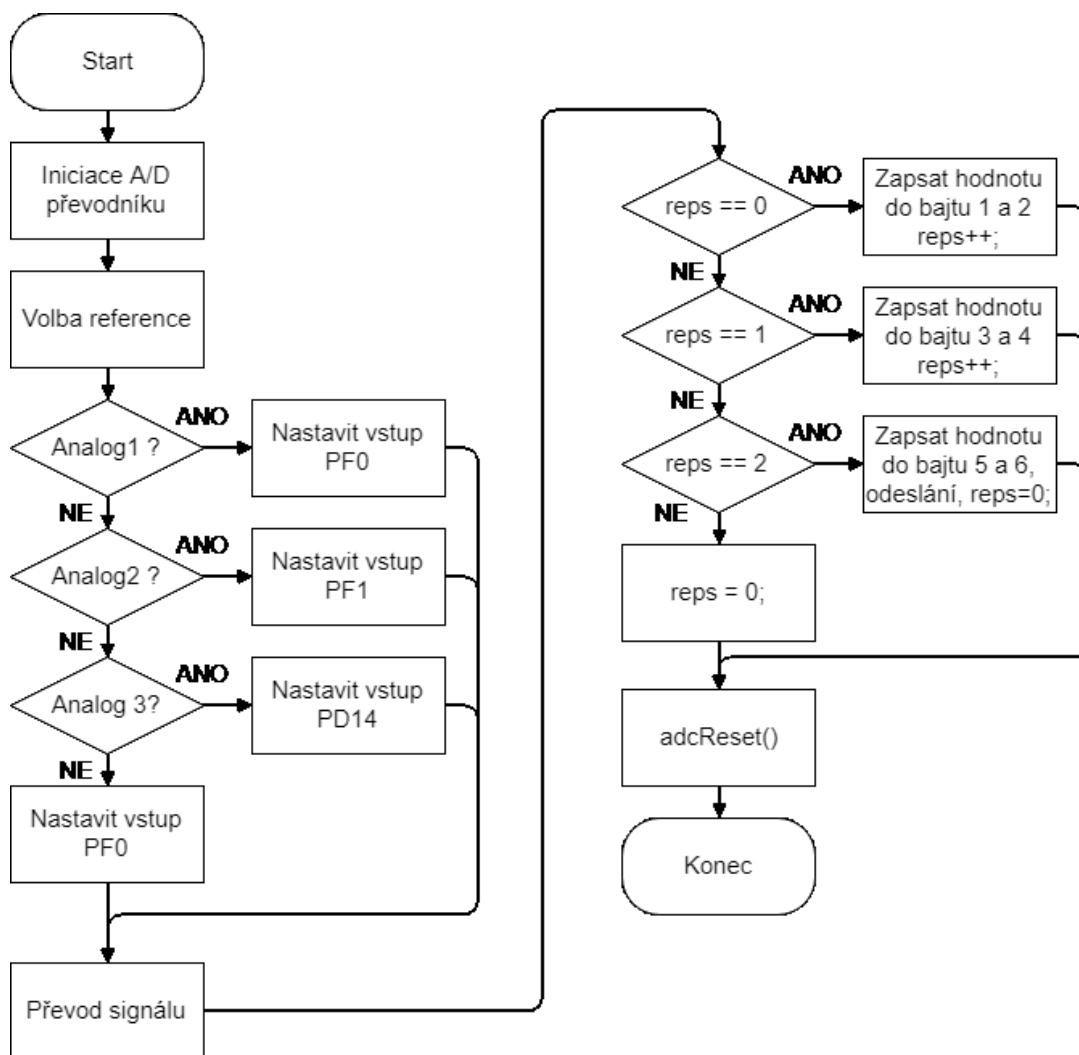
Následuje měření a převod na 16—ti bitové neznaménkové integery do formy charakteristiky. K odeslání je použit příkaz *gecko_cmd_gatt_server_send_characteristic_notification*. Posledním provedeným úkonem je *adcReset()*.

2.2.5.2 Režim *Override* a funkce pro převod jediné hodnoty

Režim *Override* (přepsání) je druhou možností pro využití charakteristiky analog. Byla navržena tak, aby postupně plnila globální proměnnou. Dojde tak k přenosu tří výsledků v jednom

paketu. Tím je umožněno navýšení maximální možné vzorkovací frekvence na trojnásobek oproti výchozí funkci.

Volba pinů a je spravována pomocí proměnné *control_point* viz podkapitola 2.2.3.3 a 2.1.3.3.



Obrázek 20: Struktura funkce režimu Override.

Pro účely této funkce je vytvořena globální osmi bitová proměnná *reps*, pomocí které je v aplikaci uloženo pořadí opakování probíhajícího měření. Při třetí iteraci je sestaven finální paket, hodnota je odeslána a *reps* je vynulována. Aby nedocházelo k započatí nového měření s hodnotou opakování větší než nula, je *reps* nulována při ukončení aktivního měření viz 2.2.3.2.

2.3 Návrh aplikace pro Android

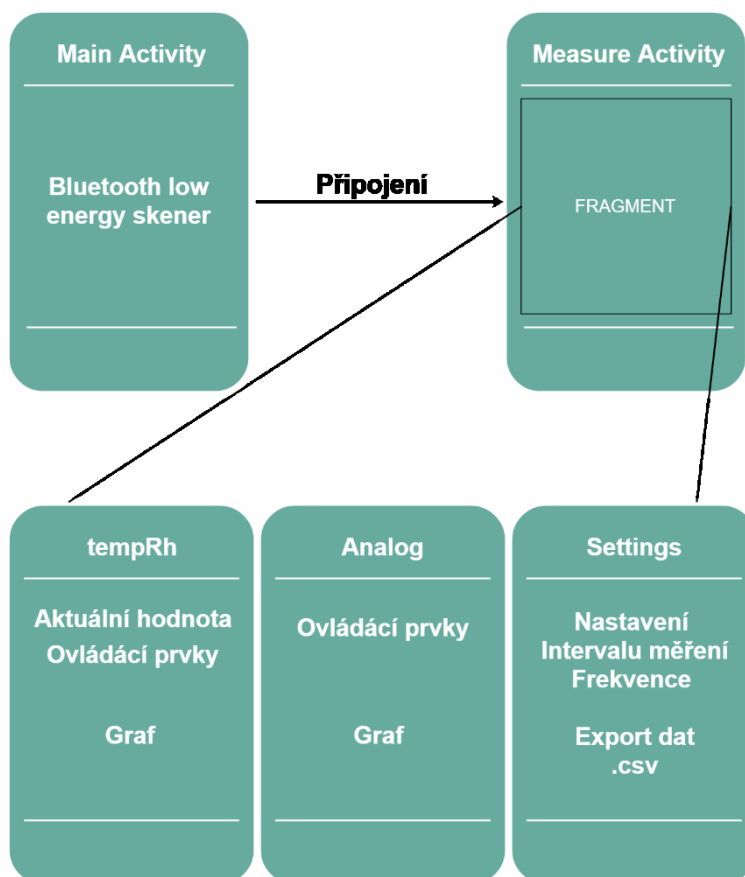
Velkou částí této práce byl také návrh aplikace pro smart zařízení. Platformou pro tento projekt byl zvolen operační systém Android. Požadavky na aplikaci byly následující:

- Nalezt Bluetooth LE zařízení a připojit se k nim,
- umožnit zapnutí měření na BGM113 a nastavení parametrů,

- vizualizace obdržených dat,
- možnost exportování výsledků do přijatelného formátu.

Aplikace by tedy měla fungovat, jako jednoduchý Bluetooth LE skener, pomocí kterého bude umožněno připojení k nalezeným zařízením a další části aplikace by měly zvládat komunikaci s BGM113. Jako vývojové prostředí bylo využito *Android Studio*. Úroveň API je 23. Zdrojový kód této části projektu je obsažen v Příloha B.

2.3.1 NÁVRH ZÁKLADNÍ STRUKTURY APLIKACE

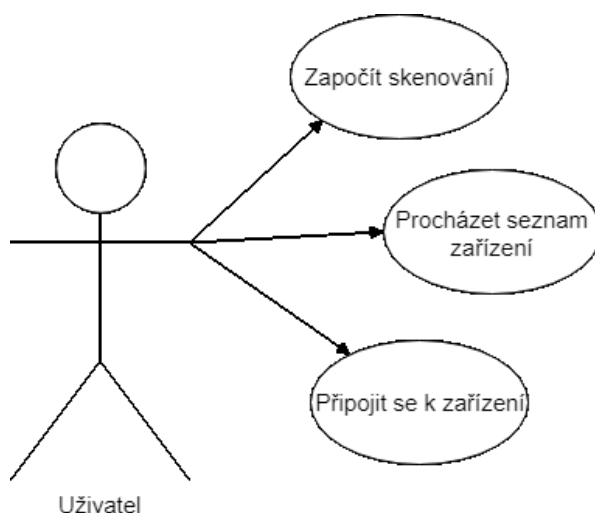


Obrázek 21: Struktura navržené aplikace.

Základním prvkem každé Android aplikace jsou aktivity. Nejprve tak byla navržena vnitřní struktura a jednotlivým aktivitám byly přiřazeny funkce. Navrženy byly dvě aktivity. První z nich obstarává vyhledání Bluetooth LE zařízení a jejich zobrazení v přehledné formě. Po zvolení podporovaného modulu dojde k přechodu do nové aktivity. Zde je spravováno spojení, příjem a ukládání dat a veškeré prvky které se týkají komunikace s BGM113, včetně exportu dat. Zároveň ale obsahuje místo pro fragmenty. S jejich pomocí je řešena veškerá vizualizace přijatých dat a zároveň obsahuje uživatelské rozhraní pro nastavení prováděných měření.

2.3.2 HLAVNÍ AKTIVITA

Jako první, po startu aplikace, je otevřena aktivita Main, v této jsou řešeny oprávnění potřebná pro funkčnost aplikace, zobrazování nalezených vysílačů Bluetooth LE. V následujících podkapitolách jsou řešeny jednotlivé obsažené funkční prvky.



Obrázek 22: Diagram užití hlavní aktivity.

2.3.2.1 Potřebná povolení pro funkčnost aplikace

Na začátku je potřeba požádat uživatele o povolení potřebných k provozu aplikace. Mezi ty patří:

- povolení týkající se Bluetooth, a to včetně přístupu k hrubé poloze uživatele, jelikož ta je pro vyhledávání také potřeba.
- Protože je implementován i export dat je nutné získat přístup k možnostem čtení a zapisování na zařízení.

Kontrola, zda je povoleno vše potřebné, je provedena při startu aplikace pomocí vytvořené metody *permissionRequest()*, ta postupně zjistí jaká povolení jsou udělena, případně si vyžádá chybějící. K této činnosti slouží metody *checkSelfPermission()* a *requestPermission()*.

Výsledky těchto žádostí jsou poté zpracovány, v případě neudělení požadovaného oprávnění je aplikace ukončena.

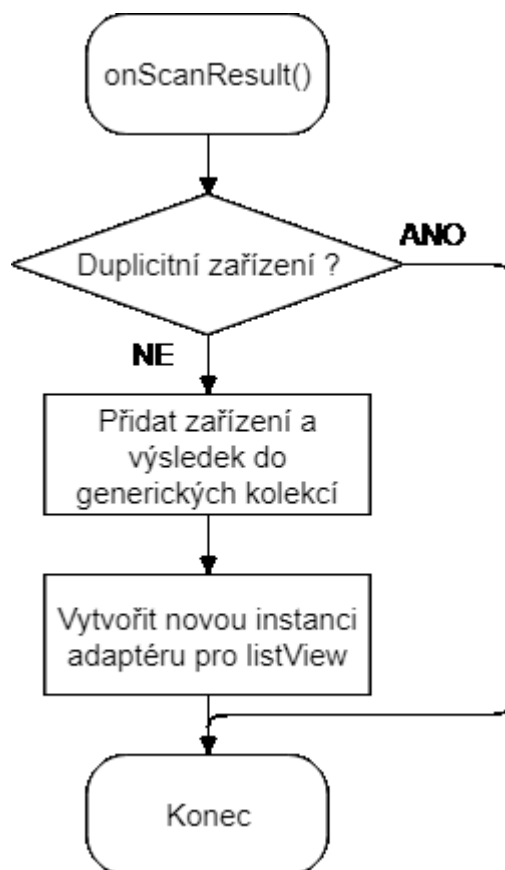
2.3.2.2 Bluetooth LE skener

Skener implementovaný v této části, byl napsán pomocí informací získaných z [15]. Dále byl upraven dle potřeb této aplikace.

Pro veškerou práci s Bluetooth bylo potřeba implementace třídy *BluetoothAdapter*. Aby bylo možné získat její instanci, je potřeba třídy *BluetoothManager*. Po úspěšné inicializaci adaptéru lze teprve vytvořit instanci třídy potřebné k nalezení Bluetooth LE zařízení, *BluetoothLeScanner*. Tato část kódu

je provedena při startu aplikace. V případě, že při startu nebyl Bluetooth na zařízení zapnut, je požádáno o jeho spuštění a předchozí rutina je provedena znovu.

Pro uložení nalezených zařízení byly vytvořeny dvě generické kolekce. První přijímá objekty typu `BluetoothDevice`, potřebné k navázání spojení. Druhá ukládá objekty `ScanResult`, pomocí kterých je tvořen seznam v uživatelském rozhraní.



Obrázek 23: Průběh metody při nalezení výsledků.

Pomocí tlačítka je spuštěno hledání okolních zařízení, které probíhá na novém vlákne. Aby nebylo možné vytvořit několik souběžně skenujících vláken, je tlačítko po dobu aktivního vyhledávání ve stavu nepovoleno. Před tímto procesem jsou ještě vyprázdněny generické kolekce, pro případ, že je vyhledávání opakované. Dále je resetován adaptér pro vizualizaci nalezených zařízení. Skenování je ukončeno buď vybráním předmětu ze seznamu zařízení nebo automaticky po pěti vteřinách.

Výsledky jsou přijímány v metodě `onScanResult()`, pomocí instance třídy `ScanCallback`. Prvně je zkontrolováno, zda nejde o duplicitní hodnotu. Následně jsou výsledky přidány do generických kolekcí a je vytvořena nová instance `myAdapter`. Tato třída je dědicem `ArrayAdapter` a slouží ke konstrukci prvku, navrženého pro zobrazování v `listView`. Ten poté zobrazuje jméno nalezeného zařízení, sílu signálu v dBm a zda je zařízení podporováno.

Pro přechod do další aktivity je potřeba zvolit jedno z podporovaných nalezených zařízení. Tímto je vytvořen je otevřena aktivita `Measure`, které je předáno zvolené zařízení.

2.3.3 AKTIVITA MEASURE

Po získání zařízení, ke kterému se má aplikace připojit je možné vstoupit do aktivity, zastřešující komunikaci s BGM113, ukládání a export naměřených výsledků. Všechny tyto činnosti běží mimo uživatelské rozhraní, ovládání dílčích funkcí je zajišťováno z fragmentů vytvořených pro tuto aktivitu.

Takovéto řešení bylo zvoleno, aby mohlo být z jedné aktivity spravováno veškeré ovládání BGM113 a zároveň vizualizaci či export dat. Obrovskou výhodou řešení je možnost měření a spravování všech prvků aplikace, lze tak najednou vykreslovat grafy pro teplotu, relativní vlhkost a tři různé analogové signály. Kdyby bylo použito více různých aktivit, připojení by muselo být obnovováno při každém přechodu mezi nimi.

Byly proto navrženy tři různé fragmenty, každý z nich má svou funkci, a komunikuje se zařízením prostřednictvím metod v Measure. V následujících podkapitolách je popsána funkčnost a řešení komunikace fragment → aktivita a naopak.

2.3.3.1 *Předávání dat mezi fragmentem a aktivitou*

Ačkoliv je fragment implementovaný jako objekt nezávislý na aktivitě. Každá jeho instance je přímo svázána s aktivitou, ve které se nachází. Díky tomuto faktu lze získat přístup k metodám, vytvořených v hlavní aktivitě. Tímto způsobem je taky řešena většina uživatelského rozhraní ve fragmentech. Proto obsahuje Measure veškeré metody potřebné ke komunikaci pomocí Bluetooth LE.

Pro cestu opačným směrem již tento postup není možný. Použitým řešením je vytvoření rozhraní pro předávání dat. To je implementováno pro každý fragment, kterému je potřeba předat data z hlavní aktivity. V této práci je užito dvou rozhraní. Jedno předává data fragmentu *tempRh* a nese hodnotu veličiny a zda jde o teplotu případně relativní vlhkost. Druhé přenáší pole obsahující přijaté hodnoty a bajt s informací o jaký režim a hodnoty se jedná.

Aby mohlo být rozhraní úspěšně využito, aplikace musí být schopna se na něj odkázat. Vytvořené fragmenty ovšem nemají své ID a jelikož jsou nahrávány všechny do jednoho okna, nelze se odkázat ani na to. Abychom mohli odesílat data i fragmentu, který není aktuálně zobrazen, je pro jejich identifikování využito pořadí, ve kterém jsou inicializovány. Proto je viz podkapitola 2.3.3.2 řešena problematika maximálního počtu načtených fragmentů mimo obrazovku. Pomocí třídy *SupportFragmentManager* a pořadí, v jakém byly vytvořeny, tak lze získat instanci fragmentu, které mají být hodnoty předány.

2.3.3.2 *Struktura aktivity Measure*

Aby mohla aktivita využívat fragmentů, musí pro to být upravena. Android Studio obsahuje předpřipravený typ *Tabbed Activity*. Ta inicializuje základní prvky, jako je upravená hlavička aplikace, umožňující přepínání aktivní záložky. Jednotlivé navržené fragmenty, se tak při správné implementaci stávají záložkami. Ty se otvírají v instanci *ViewPager*, kterému je přiřazen objekt vytvořený v rozvržení stránky. S jakou záložkou uživatel aktuálně pracuje je zjišťováno pomocí metod v též třídě.

Aby nedocházelo k nechtěnému ukončení fragmentu, nacházejícího se mimo aktuálně zobrazenou obrazovku. Musel být navýšen limit načtených fragmentů mimo obrazovku. *ViewPager*. K vytvoření plynulého přechodu mezi záložkami, udržuje načtené ty záložky, které mohou aktuálně být zobrazeny (jednu vlevo a jednu vpravo). V případě tří záložek, by tak bez navýšení limitu docházelo

k zániku prvního načteného fragmentu při navigaci v třetím (vpravo již není očekáván fragment, a pro posun vlevo je udržován pouze druhý, první je ukončen). Tím by byla ztracena aktuální nastavení uživatelského rozhraní, ale také pořadí načtení fragmentů a tím i jejich číslování. Pro lepší ilustraci této problematiky lze využít Obrázek 21, kde *tempRh* je prvním fragmentem a v případě prohlížení Settings by byl ukončen.

Číslování záložek se využívá později při řešení komunikace mezi aktivitou a fragmentem. Ten se totiž chová jako „aktivita v aktivitě“ a se svým rodičem sdílí pouze prostor v uživatelském rozhraní.

2.3.3.3 *UUID a jejich užití v aplikaci*

Veškerá komunikace smart zařízení s BGM113 vyžaduje znát UUID charakteristik, služeb a deskriptorů. Ty jsou získány při sestavení GATT a je potřeba je definovat v kódu.

Všechna UUID, která používá Bluetooth SIG jsou definována a mění se pouze 16—ti bitové, případně 32—bitové číslo. Následující vzorec popisuje jejich formu, číslo je v hexadecimální formě a x označují 16—ti bitový kód:

$$UUID = 0000\ XXXX - 0000 - 1000 - 8000 - 00805F9B34FB \quad (3)$$

Veškeré uváděné hodnoty jsou proto převedeny do úplné formy 128—bitového čísla. Teprve pomocí něj je možné se odkazovat na charakteristiky, služby, případně deskriptory.

Tabulka 5: UUID potřebné ke komunikaci, uvedeny v hex formě.

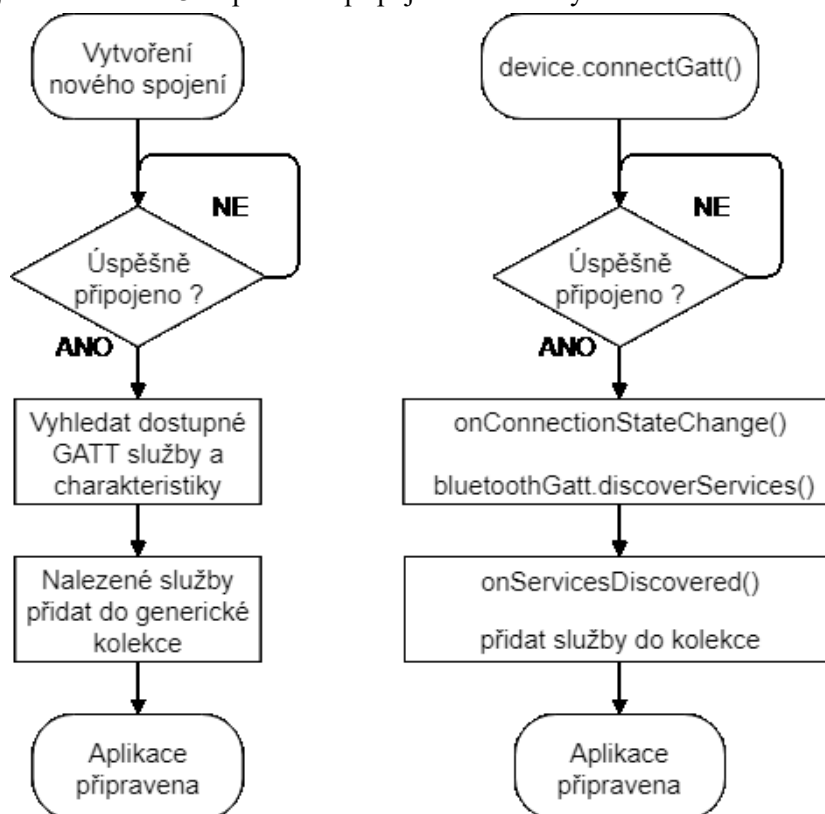
Služby	
Environmental Sensing	181A
Automation IO	1815
Charakteristiky	
Temperature	2A6E
Humidity	2A6F
Analog	2A58
User Control Point	2A9F
Frequency	5e341dfb-df04-4b13-bb0b-95138113a126
Measurement Interval	2A21
Deskriptor	
Client Characteristic Configuration	2902

2.3.3.4 *Připojení a komunikace s BGM113*

Prováděná operace se připojuje ke GATT serveru na daném zařízení. Metody, pomocí kterých probíhá komunikace s BGM113, jsou ze třídy *BluetoothGatt*.

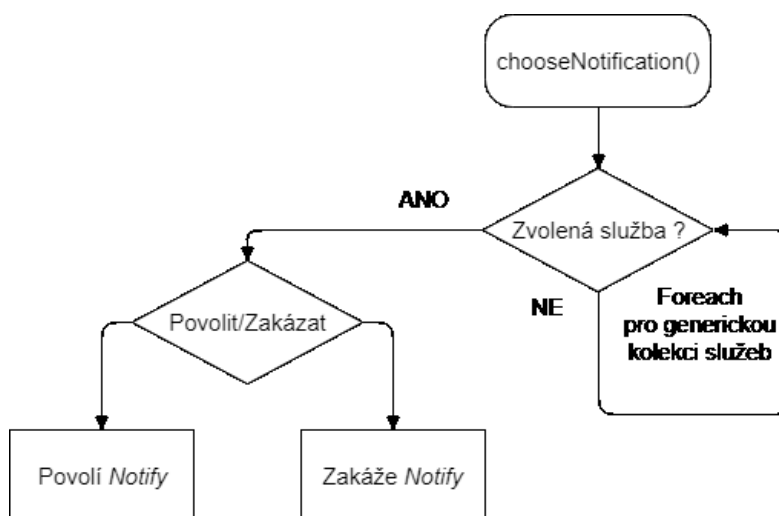
Je potřeba vytvořit instanci třídy *BluetoothDevice*, a té přiřadit hodnotu, která byla předána z hlavní aktivity (nalezené a vybrané zařízení). Poté lze vytvořit aktivní spojení, s nastavenými parametry. Důležitým parametrem je *autoConnect*, pokud je nastaven na true, zahájí při případné ztrátě

BGM113 pokusy o opětovné připojení. V případě úspěchu je ovšem potřeba nastavit veškeré parametry měření znovu, jelikož BGM113 se při ztrátě připojení vrátí do výchozího stavu viz 2.2.3.5.



Obrázek 24: Sled metod pro vytvoření nového spojení.

Postupnost událostí při novém spojení je vyobrazena na Obrázek 24. Pravá větev vyjadřuje posloupnost volaných metod a odpovědi na ně. Kde *onConnectionStateChange()* a *onServiceDiscovered()* jsou *callback* (volané po úspěšném dokončení předchozí metody). Následně je připojení udržováno, dokud není aplikace ukončena, případně dojde k navrácení do hlavní aktivity.



Obrázek 25: Průběh metody pro volbu charakteristiky.

Jakmile je spojení aktivní a rutina připojení proběhla, je možné začít komunikovat se serverem GATT. První z možností komunikace je povolení *Notify*. Po provedení tohoto úkonu začnou být přijímány naměřené hodnoty. Na tyto metody se proto odkazují tlačítka z fragmentů. Ta musí předat UUID charakteristiky a služby, a zda má být *Notify* povoleno či zakázáno. K volbě charakteristiky je použit příkaz *foreach* na služby v kolekci. Charakteristika se poté získá jednoduše metodou *getCharacteristic()*.

Pro povolení *Notify* je nutno, pomocí *BluetoothGatt* volat metodu *setCharacteristicNotification()*. Zároveň je potřeba změnit hodnotu deskriptoru s UUID hex 2902. Návaznost v kódu BGM113 je při události v kapitole 2.2.3.2, kde je tato hodnota předána pro zjištění, zda došlo k povolení či zakázání.

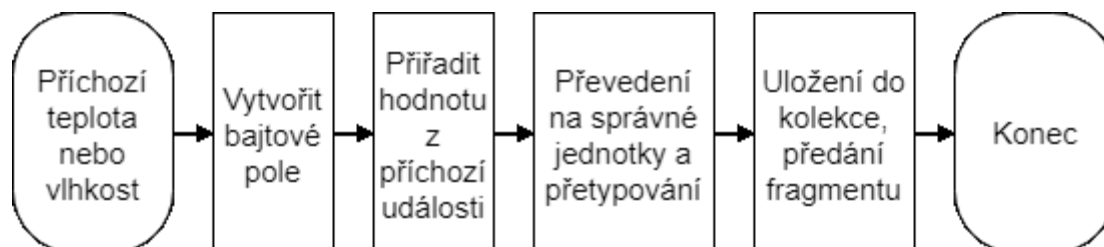
Další komunikační operací je zápis hodnoty do charakteristiky. Z fragmentu Settings získáme požadovanou frekvenci nebo interval měření. Také je potřeba připojit potřebná UUID, vytvořená metoda poté vybere charakteristiku obdobně jako na Obrázek 25. Poté je provedena typová konverze na formát definovaný GATT viz podkapitoly 2.1.2 a 2.1.3.2. Až poté je možné zapsat převedené hodnoty do serveru GATT.

Metoda pro zapsání hodnoty *control_point* je až na typové konverze stejná s předchozí popisovanou. Její hodnota je sestavena ve fragmentu Analog.

2.3.3.5 Přijímání výsledků měření

Po úspěšném nastavení parametrů a zapnutí měření, začne smart zařízení přijímat výsledky. Každé přijetí notifikace vyvolá metodu *onCharacteristicChanged()*. Zde je potřeba identifikovat jaká hodnota přišla, zapsat ji do generických kolekcí a předat ji fragmentům k vizualizaci.

V první úrovni je nově přichodící událost zařazena ke zpracování podle jejího UUID. Výsledky pro teplotu a relativní vlhkost jsou zpracovány viz Obrázek 26.



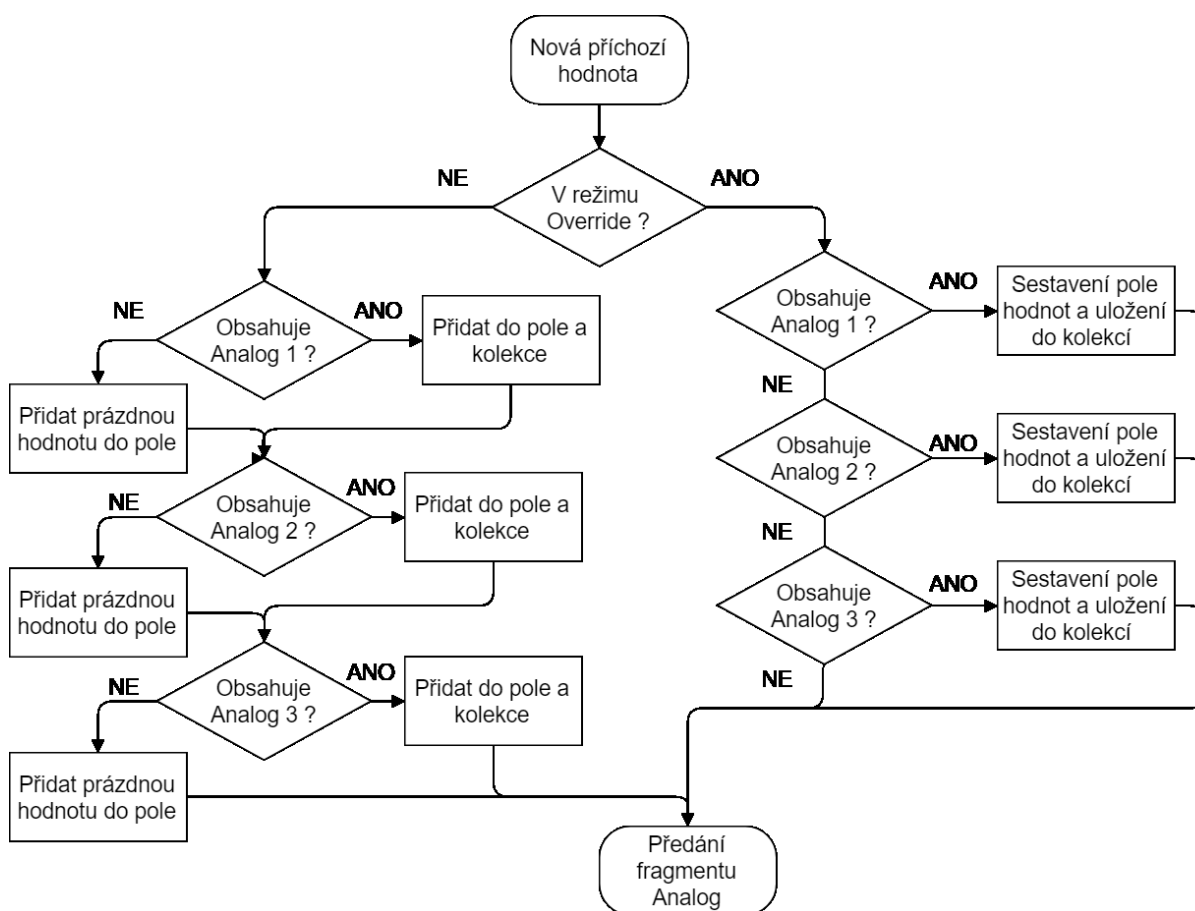
Obrázek 26: Zpracování přichodící notifikace teploty/vlhkosti.

Pro data teploty a relativní vlhkosti je provedeno zpracování zobrazené na Obrázek 26. Uložení probíhá v *Measure* aktivitě do generických kolekcí ve formátu *string*, z důvodu jednoduššího exportu do formátu *.csv* viz kapitola 2.3.3.6. Výsledky jsou předávány do fragmentu *tempRh*.

Zpracování paketu z charakteristiky *analog* je náročnější, jelikož bylo potřeba navrhnout metodu, která postupně roztřídí přichodící data a uloží je do příslušných kolekcí a zároveň je předá fragmentu *Analog*. Hodnota přichodící z BGM113 je pouze kódem, který je potřeba převést na jednotku ve voltech. Při získávání se proto nejen provede typový převod, ale i vypočítání naměřeného napětí.

Tato metoda se rozpadá pomocí příkazu *if* na dvě hlavní větve. Do jedné je vstoupeno, pokud jsou přichodící data v režimu *Override* viz kapitola 2.2.5.2 a část druhá pro případ výchozího formátu.

Každá větev postupně naplní pole hodnot, které dále předáno k vizualizaci. Vnitřní struktura je vyobrazena na Obrázek 27.



Obrázek 27: Průběh zpracování hodnot z charakteristiky Analog.

2.3.3.6 Export dat do formátu CSV

Ačkoliv jsou data vizualizovány v reálném čase, nelze zobrazovat různé části naměřených hodnot. Pro umožnění další práce mimo aplikaci, byla navržena metoda pro export. Pro její vytvoření byla využita opensource knihovna *opencsv*.

Aby bylo možné vytvořit a uložit nový soubor, je potřeba získat cestu k místu jeho plánovaného umístění. Navržené řešení ukládá vyexportované soubory do složky dokumenty na primárním uložišti. Jméno lze upravit manuálně uvnitř fragmentu Settings. Po sestavení cesty je možné započít export. Nejdříve je vytvořen soubor, do kterého se budou zapisovat hodnoty ze zvolené kolekce. Poté dojde ke kontrole, zda vytvoření souboru proběhlo úspěšně. Dále dojde k postupnému vepsání, naměřené hodnoty z kolekce jsou ukládány po řádcích. Kód týkající se zápisu musí být ohraničen blokem *try—catch* pro případ, že selže vytvoření souboru nebo zápis samotný.

2.3.4 FRAGMENT TEMPRH

Vizualizace dat je obstarávána fragmenty, pro teplotu a relativní vlhkost slouží tempRh. Ten obsahuje dvě textová pole pro zobrazení aktuální naměřené hodnoty, tlačítko *switch* pro spuštění měření

a graf pro zobrazení hodnot v reálném čase. *GraphView* je opensource knihovna využitá k vykreslení naměřených hodnot, je využívána, protože android neobsahuje výchozí knihovny pro grafy.

Při spuštění jsou inicializovány grafy, aby bylo možné zobrazit 10 posledních hodnot naměřené teploty a relativní vlhkosti. Jelikož je minimální interval měření jedna vteřina, pro aktualizaci grafu byla zvolena metoda *appendData()*. Ta přidá do série jeden bod a posune graf na konec osy x. Takto je vytvořen graf, který se aktualizuje pokaždé, co obdrží nové hodnoty.

Jediný ovládací prvek volá metodu *chooseNotification()* viz 2.3.3.4, která spouští měření. Po jeho spuštění začne hlavní aktivita přijímat a ukládat data, která jsou následně předána tomuto fragmentu. Tento ovládací prvek povolí nejprve *Notify* pro relativní vlhkost, jelikož ta nespustí vnitřní časovač na BGM113 viz 2.2.3.2. Následně je vyčkáno 200 ms a až poté je povolena charakteristika pro teplotu. Tento interval je musí být využit, jelikož příkazy, kterými jsou povolovány notifikace, vyžadují *Acknowledgement*. To v praxi znamená, že nelze odeslat dva příkazy najednou, ale je potřeba vyčkat na potvrzení a následně odeslat další. V případě porušení tohoto principu je druhý zaslaný příkaz ignorován.

2.3.5 FRAGMENT ANALOG

V případě analogových signálů byla potřeba složitější implementace fragmentu. Musí totiž zvládat volbu pinu pro A/D převodník, povolení režimu *Override* a spuštění měření. Na pozadí jsou poté přijímány naměřené hodnoty a zároveň je vykreslován graf v reálném čase.

Pro povolení pinů byly použity tlačítka typu *CheckBox*, díky nim má uživatel přehled o povolených vstupech. Režim *Override* a spuštění měření je spravováno dalšími samostatnými tlačítky.

Pro normální měření dojde k povolení analogového vstupu, což volá adresující metodu v aktivitě *Measure* a zapíše hodnotu viz 2.3.3.4. Následně je povoleno měření.

Tabulka 6: Hodnoty využitě pro nastavení User Control Point.

Nastavení User Control Point	Povolení (hex)	Zakázání (hex)
Analog 1	81	7E
Analog 2	82	7D
Analog 3	84	7B
Override	0 a 90	6F

Hodnoty v Tabulka 6 byly vybrány podle 2.1.3.3, hodnota 0 značí potřebu prvně vynulovat veškeré předchozí nastavení a následné povolení *Override*. Aby nebylo umožněno povolit více vstupů pro *Override*, jsou tlačítka po volbě jednoho vstupu zakázána do doby, než je zvolený vstup zakázán. Tlačítka tak předávají hodnoty, pomocí kterých je ovládáno měření na BGM113.

Dalším řešeným problémem bylo zobrazování dat do grafu. Jelikož při měření získáváme vzorky tří signálů s frekvencí až 150 Hz, není možné využít stejné metody k aktualizaci grafu, jako u tempRh. Byla proto navržena metoda, která obnovuje graf samostatně v intervalu 500 ms. Z tohoto důvodu byl implementován časovač formou třídy *Handler*. V něm jsou do grafu nahrávány hodnoty, které byly uživatelem povoleny k měření.

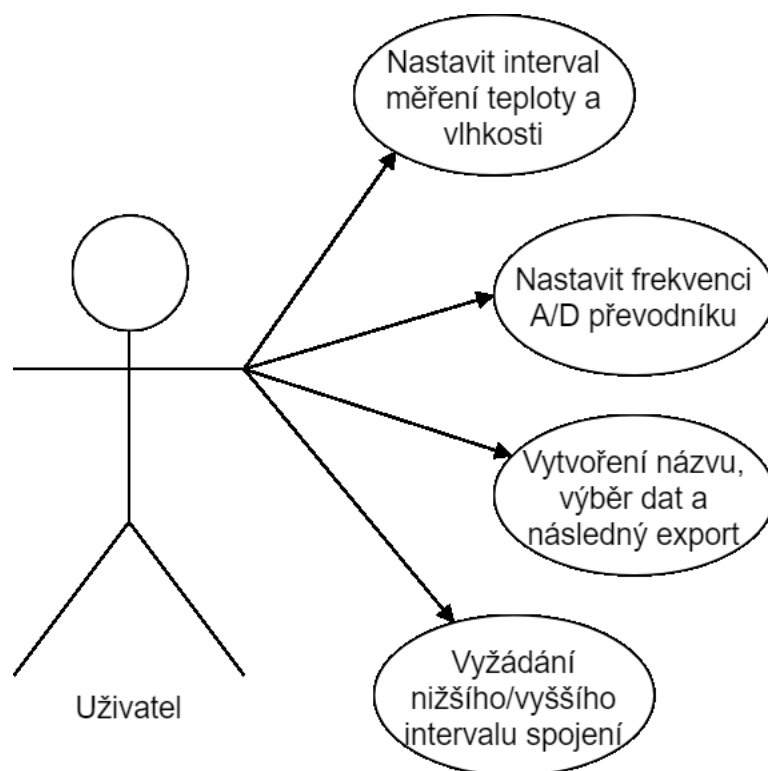
Aby mohly být hodnoty úspěšně resetovány v grafu, knihovna *GraphView* obsahuje metodu *resetData()*. Ta vyprázdní aktuální graf a naplní jej polem hodnot. Jelikož tyto metody běží

v samostatném vlákne, musí být řešena synchronizace. K hodnotě generických kolekcí, totiž přistupují metody zapisující nové příchozí hodnoty, a zároveň co 500 ms jsou přečteny a zobrazeny do grafu. Tato problematika byla vyřešena implementací třídy *ReentrantLock*. Při čtení jsou tak kolekce uzamčeny a po dokončení potřebných operací opět odemčeny.

Dále byla vytvořena metoda pro přibližování a oddalování v rámci osy x. Tohoto bylo docíleno změnou počtu zobrazovaných hodnot v grafu.

Třídění příchozích hodnot probíhá podobně jako na Obrázek 27, rozdíl spočívá v přidělování x-ových souřadnic, aby mohl být vytvořen formát *DataPoint*, potřebný pro aktualizaci grafu.

2.3.6 FRAGMENT SETTINGS



Obrázek 28: Diagram užití fragmentu Settings.

Fragment Settings byl navržen jako část aplikace, která umožňuje nastavit další dílčí části měření, spravovat spojení a export naměřených výsledků. Funguje spíše jako prvek uživatelského rozhraní, protože neobsahuje téměř žádné vlastní metody. Prvky použité uvnitř fragmentu se odkazují na předdefinované funkce viz 2.3.3.4 a 2.3.3.6. Obrázek 28 popisuje, jaké operace lze z této části provést.

Vyžádání změny intervalu spojení je uskutečněno pomocí metody *requestConnectionPriority()*. Tato možnost ovšem není doporučena, jelikož při testovacích měření docházelo při změně intervalu ke ztrátám spojení. Výchozí hodnota tohoto nastavení, je proto z pohledu hladké funkčnosti aplikace nejvhodnější.

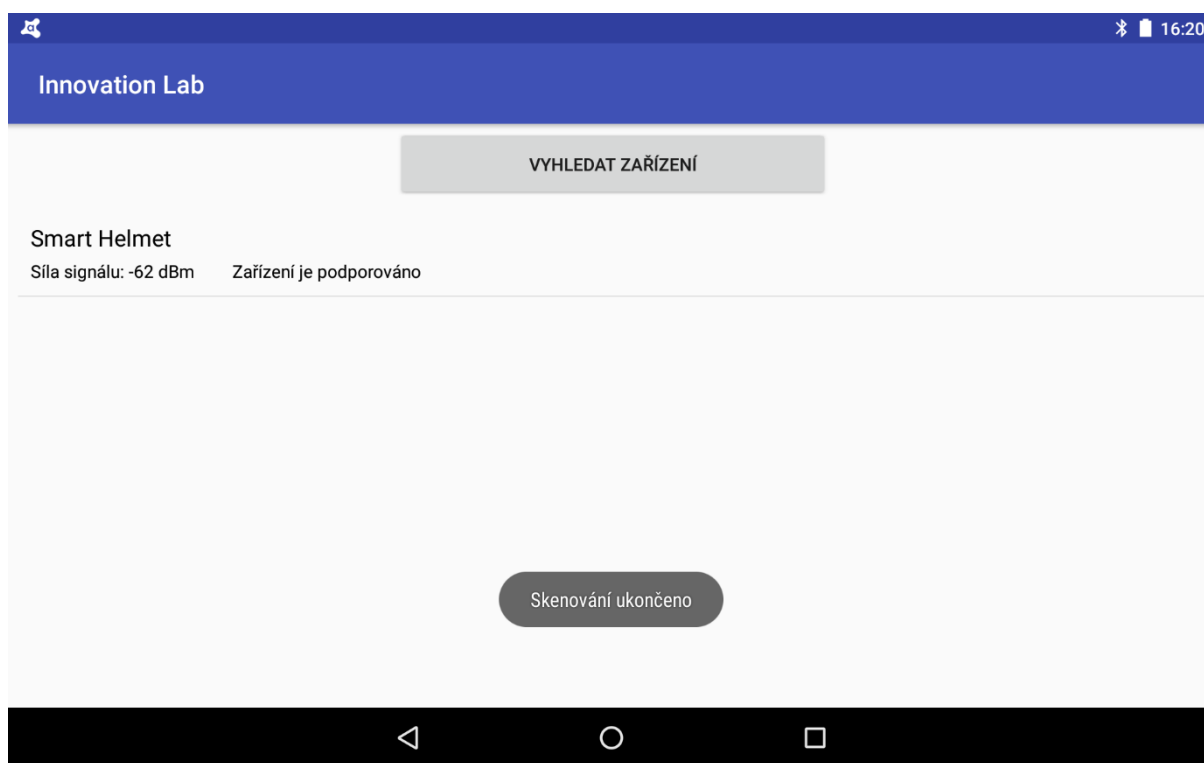
2.4 Výsledky měření a prezentace uživatelského rozhraní

V následujících podkapitolách jsou shrnuty výsledky měření a energetické profily. Také je zobrazeno uživatelské rozhraní vytvořené aplikace.

2.4.1 UŽIVATELSKÉ ROZHRAŇÍ ANDROID

Pro účely aplikace bylo navrženo uživatelské rozhraní. Postupně jsou zobrazeny a popsány navržené části Android aplikace. Každý snímek představuje aktivní grafické uživatelské rozhraní (GUI). Uživatelské rozhraní funguje jak ve vertikálním, tak v horizontálním módu. Pokud ovšem dojde k otočení obrazovky v aktivitě Measure, dojde ke ztracení spojení a zároveň znovu načtení fragmentů. Ty mohou být načteny v jiném pořadí a aplikace tak nefunguje. Je proto doporučeno používat aplikaci s vypnutým automatickým přetáčením.

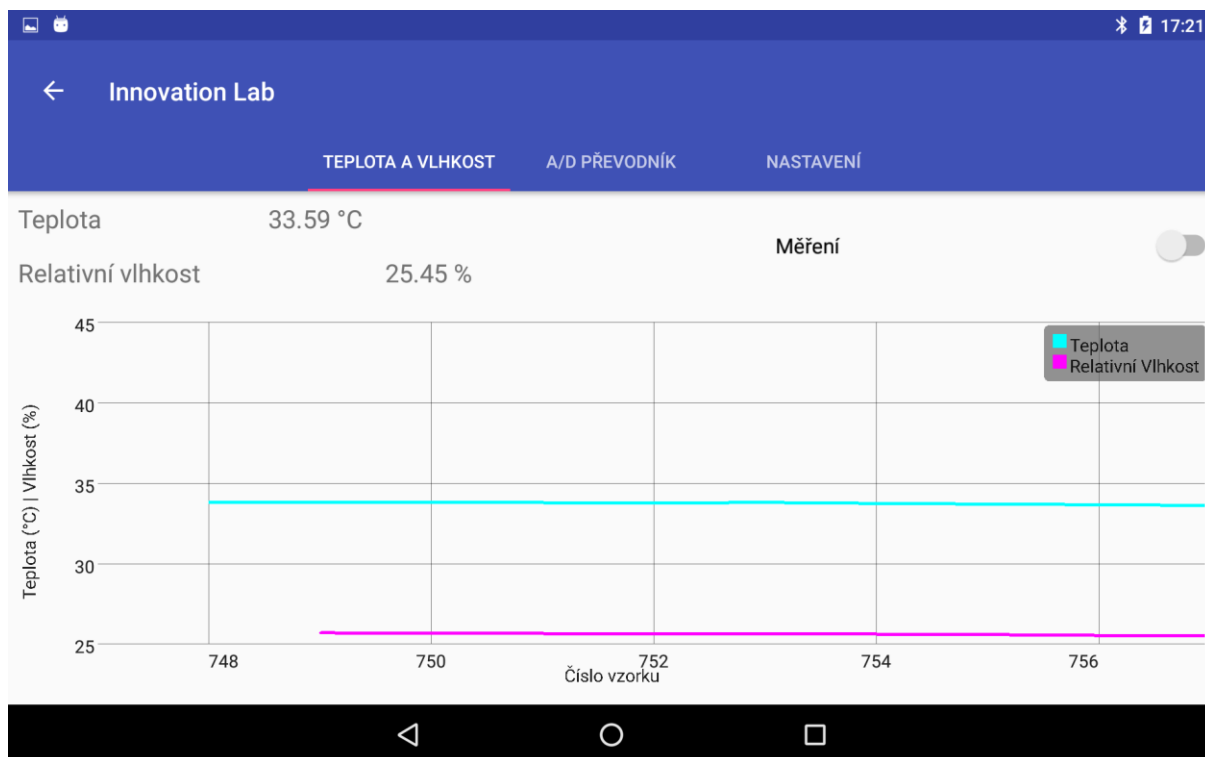
2.4.1.1 Aktivita MainActivity – Bluetooth LE skener



Obrázek 29: Hlavní aktivita, Bluetooth LE skener.

Aktivita MainActivity je tvořena jedním tlačítkem, které spustí skenování Bluetooth LE. Výsledky se zobrazují pod sebe do *ListView*. Pokud poté dojde ke kliknutí na podporované zařízení, dojde k přechodu do aktivity Measure.

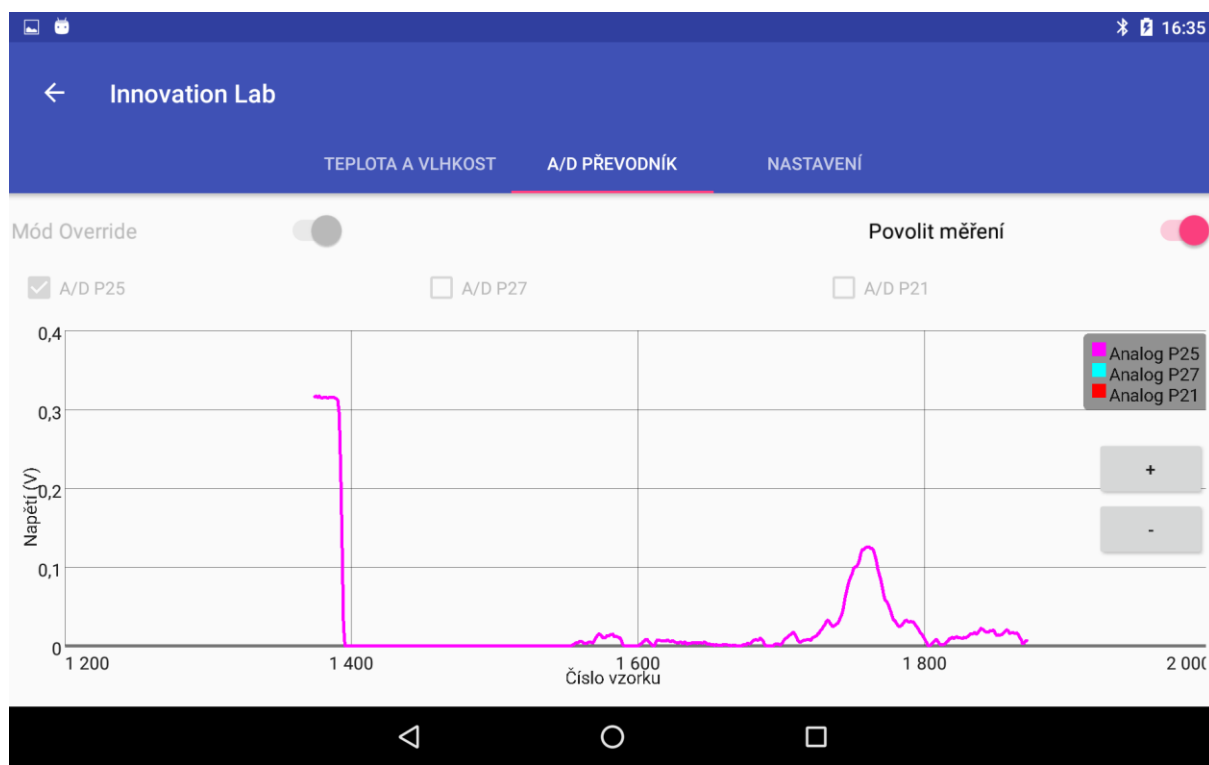
2.4.1.2 GUI fragmentu tempRh



Obrázek 30: Obrazovka s fragmentem tempRh.

Uživatelské rozhraní aktivity Measure zahrnuje krok zpět do MainActivity a výběr přechodu na určitou záložku. Zbýlý obsah rozhraní je součástí fragmentu tempRh. Tlačítkem je započato měření s předem nastaveným intervalem.

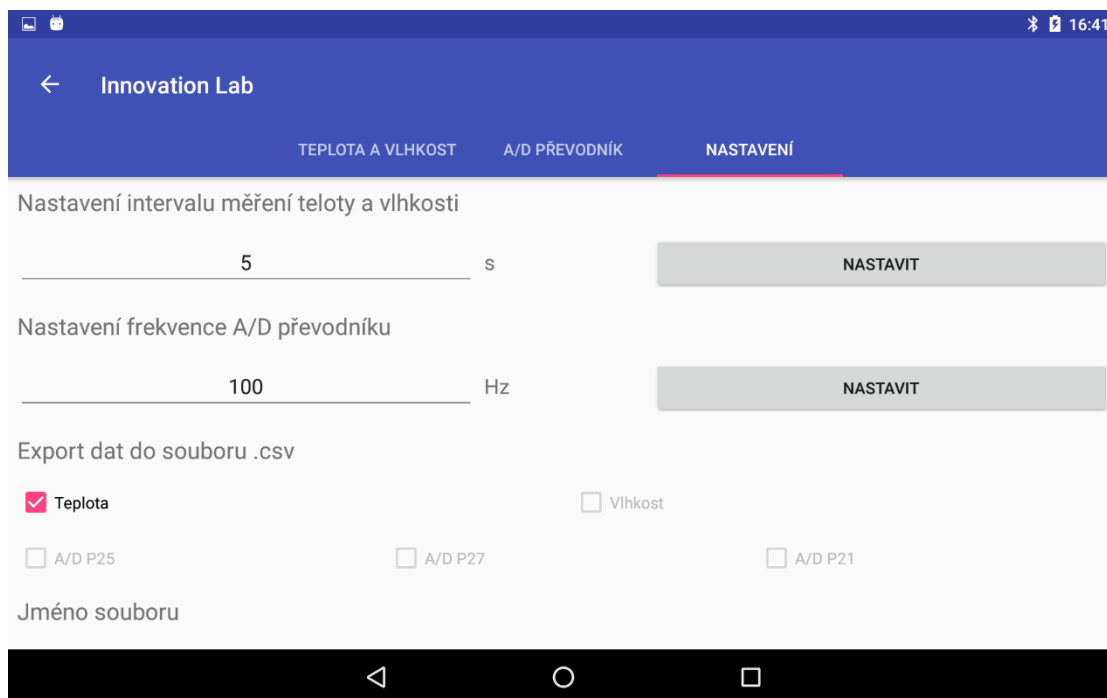
2.4.1.3 GUI fragmentu Analog



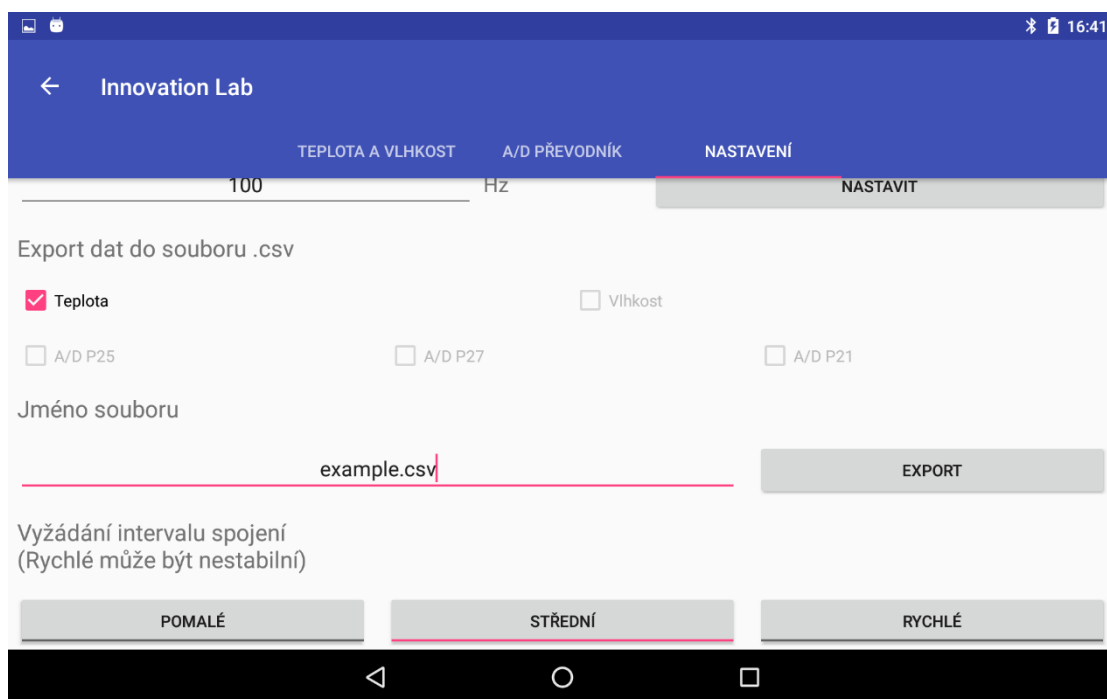
Obrázek 31: Zobrazení pracujícího fragmentu analog.

Fragment Analog obsahuje graf pro zobrazení měřených signálů, nad ním jsou prvky grafického rozhraní ovládající Bluetooth modul. Navolit v nich chybnou kombinaci není umožněno. V grafu je možné současně sledovat až tři vzorkované signály najednou, vyšší frekvence v tomto případě ale mohou způsobovat zpomalení systému z důvodu množství zobrazovaných bodů. Tlačítkem + a – lze přiblížit a oddálit sledovaný signál.

2.4.1.4 GUI fragmentu Settings



Obrázek 33: Uživatelské rozhraní fragmentu Settings.



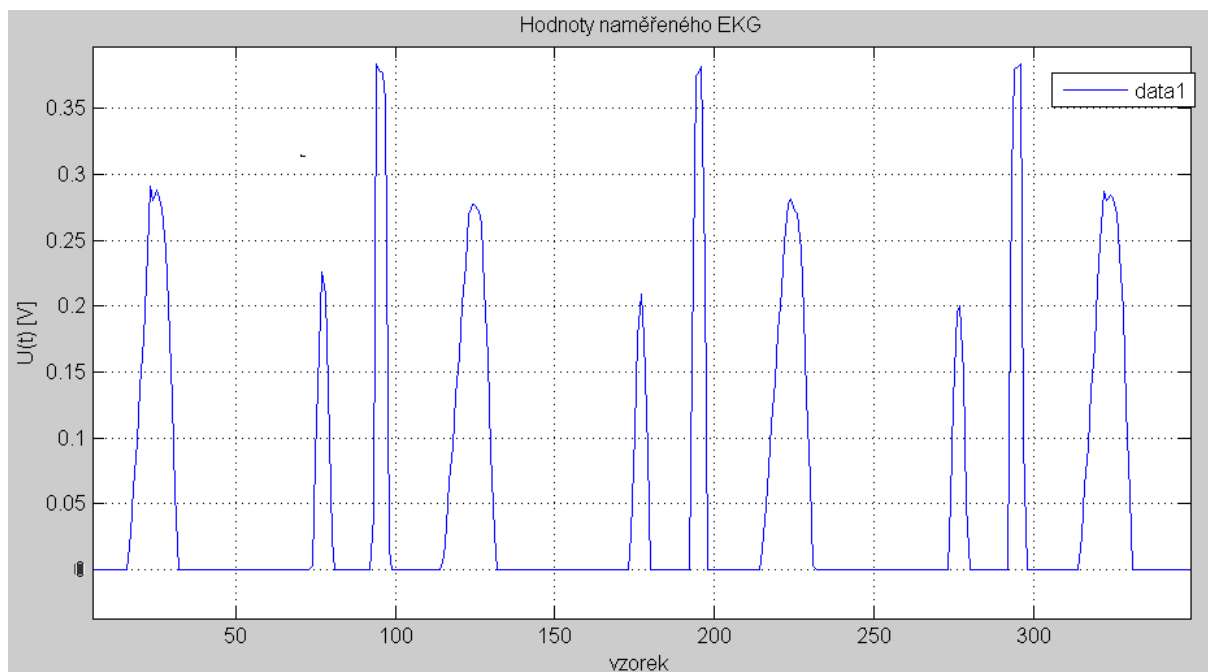
Obrázek 32: Další část fragmentu Settings.

Fragment Settings obsahuje další zbylé možnosti pro nastavení komunikace s BGM113. Frekvenci a interval měření lze přepsat viz Obrázek 33, při exportu lze vybrat jaká data (pouze jedny pro jeden export) a pojmenovat soubor. A na závěr interval spojení, viz Obrázek 32.

2.4.2 NAMĚŘENÉ ANALOGOVÉ SIGNÁLY

Pomocí zařízení byly nakonec naměřeny analogové signály EKG. Vyexportované hodnoty byly zobrazeny v programu *Matlab*.

2.4.2.1 Signál EKG



Obrázek 34: Naměřené EKG.

Obrázek 34 je výstřížkem ze série naměřených hodnot EKG. Data byly měřeny pomocí kitu EKG PhysioLab 2.4. Signál z tohoto kitu je v rozpětí -2,5 V až 2,5 V. Záporná část signálu proto nemohla být pomocí BGM113 naměřena. Vzorkovací frekvence tohoto signálu je 100 Hz.

Jelikož je ale EKG měřeno hlavně z důvodu detekce srdečního rytmu, není záporné části EKG křivky potřeba. Tep lze detekovat pomocí R vlny, ta je v signálu jasně definovaná. Měření má proto požadovanou výpovědní hodnotu.

2.4.2.2 Příklad měření zvolených analogových signálů



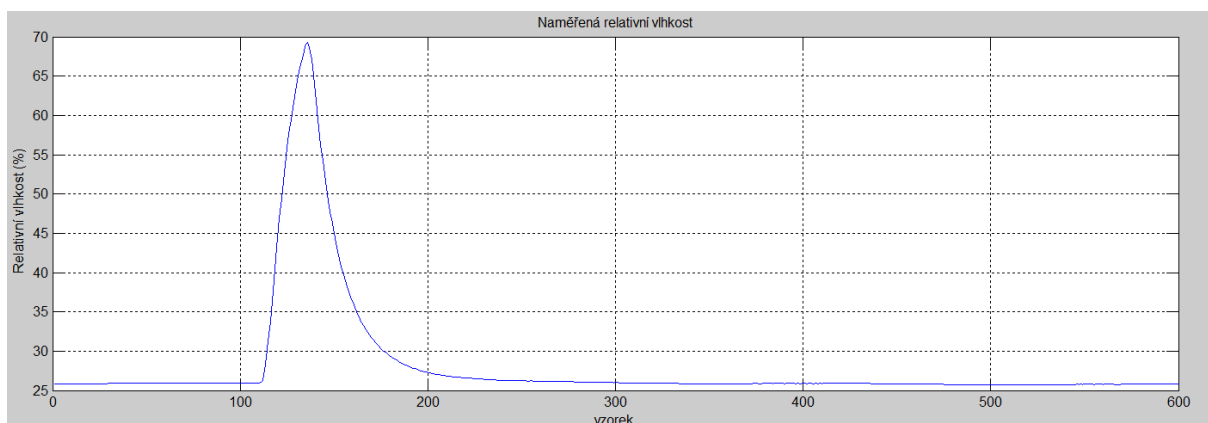
Obrázek 35: Měření zvolených analogových singálů.

Pro prezentaci výsledků měření A/D převodníku byly zvoleny dva analogové signály. Jeden trojúhelníkový (10 Hz, offset 1,5 V, amplituda 0,5 V) a druhý sinusový (10 Hz, offset 750 mV, amplituda 125 mV). Pomocí A/D převodníku byly měřeny frekvencí 100 Hz. Na Obrázek 35 je zobrazen průběh signálu v uživatelském rozhraní. Aplikace je zvládne zpracovat přehledně. Během jednoho měření lze zobrazit až tři průběhy signálů v reálném čase.

2.4.3 NAMĚŘENÉ ENVIROMENTÁLNÍ VELIČINY

Pomocí digitálního čidla Si 7021 umístěného na vývojovém kitu byla naměřena teplota a relativní vlhkost okolí. Výsledky jsou zobrazeny v programu *Matlab*.

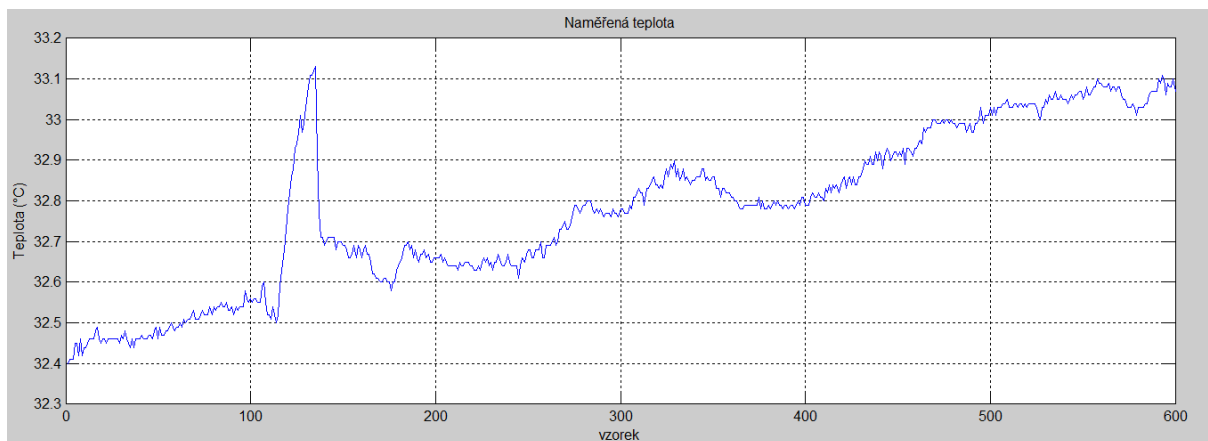
2.4.3.1 Relativní vlhkost



Obrázek 36: Naměřená relativní vlhkost.

Výsledky zobrazené na Obrázek 36 jsou z 10—ti minutového měření. Velký skok v průběhu měření byl způsoben dotykem čidla. Relativně nízká vlhkost je způsobena teplotou vývojového kitu, který se během své činnosti zahřívá.

2.4.3.2 Teplota



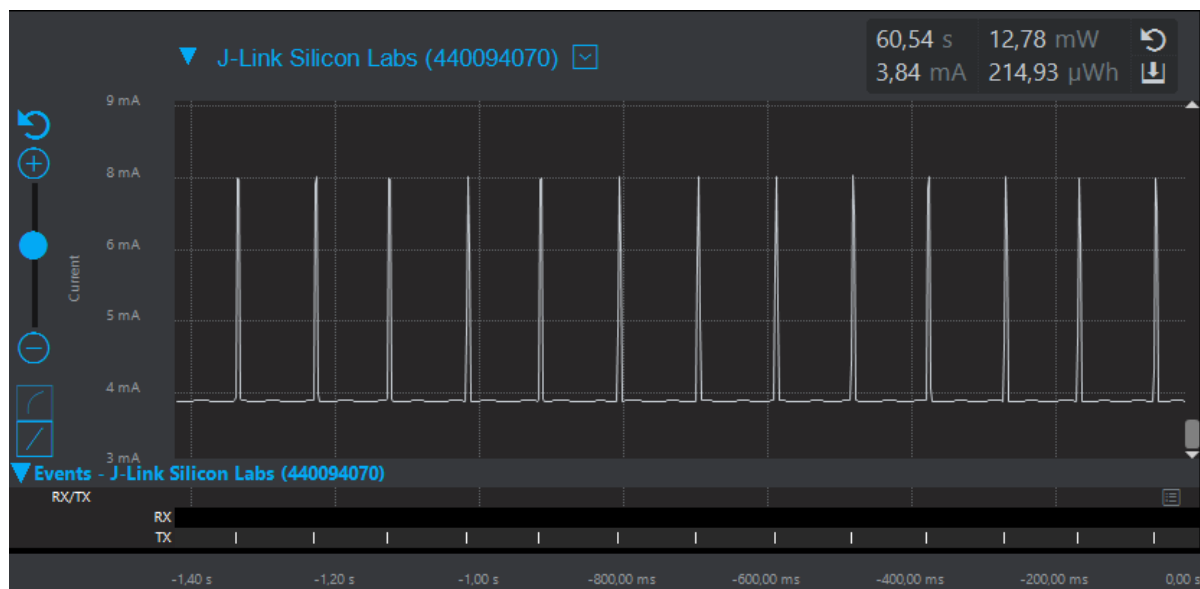
Obrázek 37: Naměřená teplota.

Výsledky z měření teploty na Obrázek 37 byly naměřeny při stejné instanci, jako výsledky relativní vlhkosti. Z výsledného grafu lze vyčíst postupné zahřívání vývojového kitu při své funkci. Skokové hodnoty byly způsobeny dotykem senzoru a korelují s hodnotami z 2.4.3.1.

2.4.4 ENERGETICKÉ PROFILY BGM113

K zjištění spotřeby zařízení bylo využito nástroje *Energy Profiler*. Ten je součástí vývojového prostředí *Simplicity Studio v4*. Následující hodnoty byly naměřeny vždy v příslušném režimu. Pro zjištění průměrné spotřeby bylo zařízení udržováno v tomto režimu po 60 sekund.

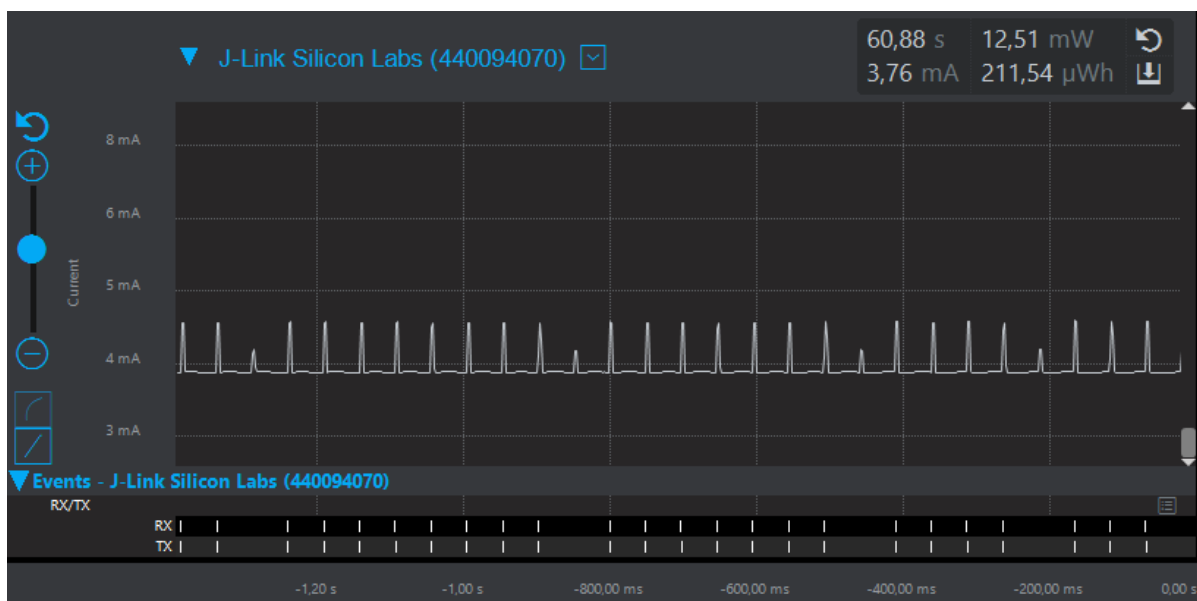
2.4.4.1 Profil pro Advertising



Obrázek 38: Energetický profil průběhu Advertisingu.

Test spotřeby byl uskutečněn pro *Advertising*. Průměrná spotřeba zařízení dosáhla 3,83 mA. Skoky ve spotřebě jsou způsobeny odesláním paketu pro *Advertising*. Jelikož délka tohoto paketu je 31 bajtů, dochází k vyšší spotřebě na Bluetooth rádiu. *Advertising* probíhá v intervalech 100 ms viz 2.2.3.1. Jeho nastavením na nižší hodnotu lze dosáhnout snížení spotřeby.

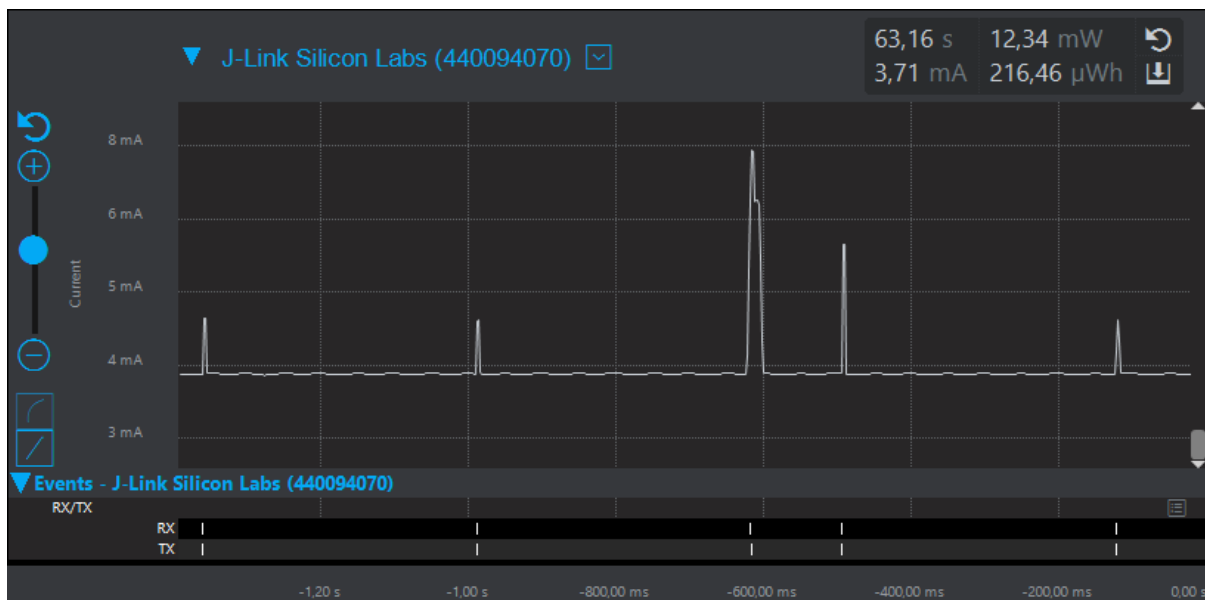
2.4.4.2 Profil připojení v klidu (bez zapnutých měření)



Obrázek 39: Profil připojení za klidu.

Po připojení spotřeba modulu klesla o 0,1 mA, to je způsobeno ukončením odesílání paketů. Namísto toho je nyní v intervalu přibližně 39 ms kontrolováno, zda nebyly obdrženy nové pakety. Do tohoto stavu BGM113 vstoupí po připojení.

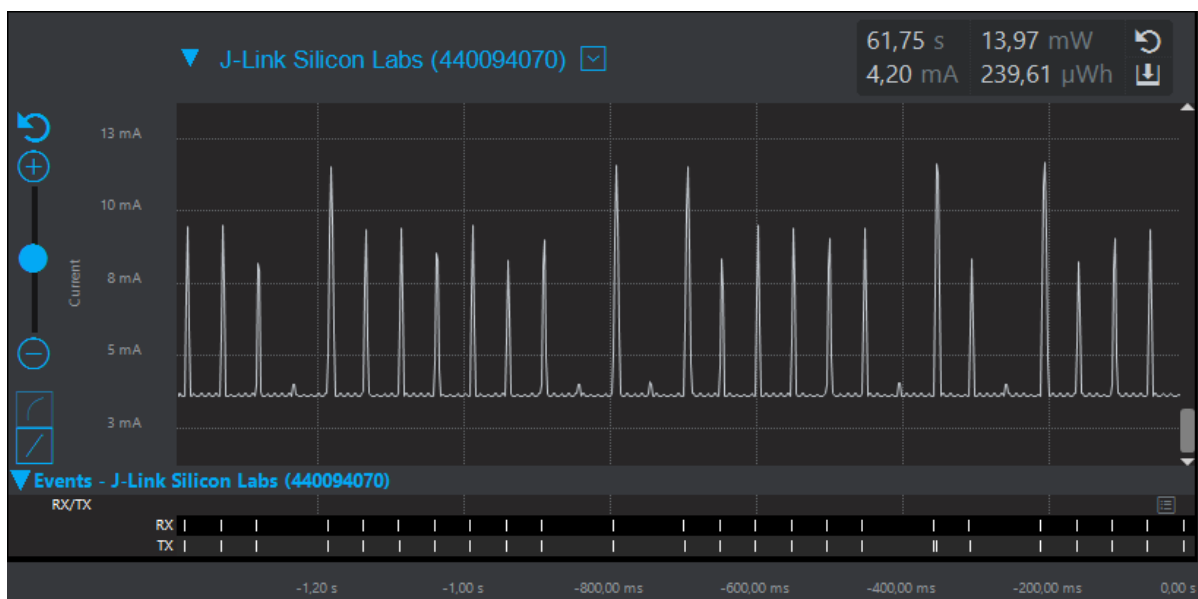
2.4.4.3 Profil měření teploty a relativní vlhkosti.



Obrázek 40: Profil měření teploty a relativní vlhkosti.

Teplota a vlhkost okolí je BGM113 měřena v intervalu pěti vteřin, to je znázorněno skokovým nárůstem spotřeby energie. Dále bylo použito nastavení intervalu spojení na pomalé. Na Obrázek 40 je zobrazen energetický profil při těchto podmínkách. Průměrná spotřeba proudu za dobu 60 sekund byla 3,71 mA.

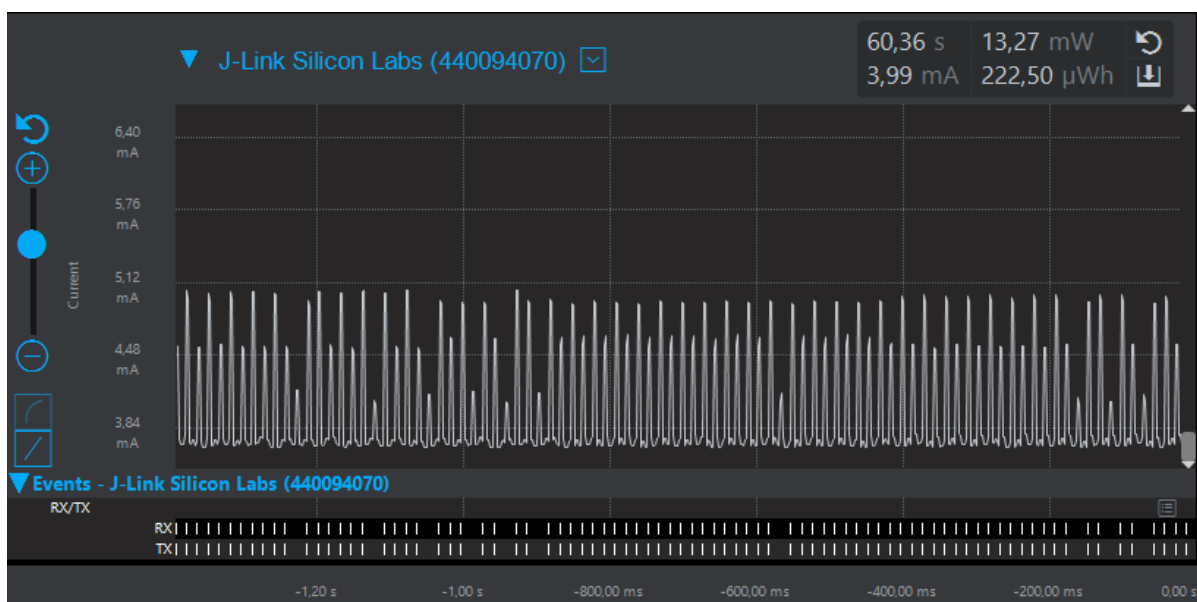
2.4.4.4 Profil měření analogového signálu při vzorkovací frekvenci 100 Hz



Obrázek 41: Profil měření analogového signálu při vzorkovací frekvenci 100 Hz.

V této části byla testována spotřeba při nastavení následujících parametrů měření: interval spojení → střední, vzorkovací frekvence A/D převodníku ve výchozím režimu je 100 Hz. Při tomto nastavení dosáhlo zařízení v průběhu jedné minuty průměrné spotřeby proudu 4,20 mA.

2.4.4.5 Profil při měření analogového signálu v režimu *Override* frekvencí 200 Hz.



Obrázek 42: Režim *Override*, vzorkovací frekvence 200 Hz interval spojení rychlý.

V této části byly nastaveny následující parametry: režim *Override*, vzorkovací frekvence 200 Hz a interval spojení na rychlý. V této části došlo k poklesu spotřeby oproti případu 2.4.4.5. To z důvodu, že režim *Override* odesílá pouze každý třetí vzorek, v některých intervalech spojení proto nedošlo k odeslání žádné informace.

2.4.4.6 Porovnání naměřených hodnot

Tabulka 7: Porovnání výsledků měření, doba měření přibližně 60 s.

	Průměrná spotřeba proudu	Spotřeba energie
Advertising	3,84 mA	214,93 μ Wh
Připojení v klidu	3,76 mA	211,54 μ Wh
Měření teploty a vlhkosti	3,71 mA	216,46 μ Wh
Vzorkování 150 Hz	4,20 mA	239,61 μ Wh
Vzorkování 200 Hz <i>Override</i>	3,99 mA	222,50 μ Wh

Všechny měření byly prováděny po dobu přibližně jedné minuty. Postupně byly proměřeny různé případy užití v aplikaci. Nejnižší spotřeby BGM113 dosahuje při nastavení delšího intervalu měření. V tomto případě bylo ještě využito digitálního senzoru k měření teploty a relativní vlhkosti. Aby BGM113 dosáhlo ještě nižší spotřeby bylo by potřeba povolit *sleep_flags* viz 2.2.2. Jelikož interval pro *advertising* je nastaven na 100 ms, Dochází při tomto profilu k vyšší spotřebě způsobené odesláním delších paketů. Nejvyšší hodnoty spotřeby 4,20 mA bylo dosaženo při měření A/D převodníkem se vzorkovací frekvencí 100 Hz a intervalem spojení nastaveným na střední.

Závěr

Teoretická část práce nastíní technologii Bluetooth low energy, aspekty výběru a hlavní rozdíly mezi zvolenými Bluetooth Smart moduly a popis konkrétního modulu Blue Gecko Module Wireless Starter Kit, jenž byl použit k uskutečnění měření, jehož výsledky jsou přiblíženy v praktické části, čímž bylo umožněno měření tří analogových signálů, teploty a relativní vlhkosti v zapojení s digitálním senzorem Si 7021-A20.

Praktická část už je detailním popisem navržených řešení a výsledků. V první řadě GATT, je tvořen jedním profilem, jenž obsahuje veškeré služby. Dvě z nich (Generic Access a Device Information) jsou určeny k identifikaci a kategorizaci zařízení. Následující dvě (Environmental Sensing a Automation IO) se týkají odesílání naměřených výsledků a možnosti nastavení parametrů měření. Jako unikátní charakteristiky byly navrženy *Analog* a *User Control Point*. *Analog* umožňuje společný přenos hodnot až tří analogových signálů, a v případě režimu *Override* tří postupně naměřených a zapsaných výsledků. Díky tomuto řešení je navýšena maximální vzorkovací frekvence, se kterou lze zobrazovat data z A/D převodníku na smart zařízení, až na trojnásobek oproti použití původní neupravené charakteristiky. Dále bylo popsáno řešení ovládání mikrokontroleru pomocí speciální charakteristiky *User Control Point*. K vytvoření GATT bylo využito grafického rozhraní, jenž je součástí vývojového prostředí *Simplicity Studio v4*.

Další tematický úsek se věnuje navrženému firmware pro BGM113, jeho struktuře a vytvořeným funkcím. Byla vytvořena konstrukce pro třídění událostí, které podle svého druhu vyvolávají příslušné funkce a tím tvoří hlavní tělo firmware. Navržená funkce pro měření teploty využívá rozhraní I²C ke komunikaci s digitálním senzorem. Byla upravena pro umožnění měření teploty a relativní vlhkosti při jednom měření senzoru. Účelem je získání obou hodnot a následné odeslání v jedné události. Následně byly přizpůsobeny dvě funkce pro A/D převodník. Pomocí výchozí funkce lze měřit až tři analogové signály v jednom momentu a následně je předat chytrému zařízení k vizualizaci, zpracování a exportu. Režim *Override* využije původně navržené charakteristiky analog, kde zapisuje namísto tří různých hodnot postupně vzorky naměřeného signálu, s každým třetím je odeslána charakteristika ke zpracování. Takto je umožněno využít až trojnásobné vzorkovací frekvence k měření analogového signálu.

K chytrému zařízení s operačním systémem Android byla navržena aplikace, jejímž účelem bylo vytvořit uživatelské rozhraní, nalézt a připojit se k Bluetooth modulu, ovládat jednotlivé parametry měření a provést vizualizaci přijímaných měření včetně možnosti exportu. Pro účely aplikace byl vytvořen *Bluetooth LE* skener, který zobrazuje jméno zařízení, sílu signálu a zda je kompatibilní s aplikací. Další část aplikace funguje na principu využití fragmentů. S jejich pomocí byla vytvořena část aplikace obsahující tři záložky pro specifické funkce aplikace. Během fungování těchto fragmentů je na pozadí udržováno *Bluetooth LE* připojení, zpracování a ukládání dat. První záložka se věnuje zapnutí a vizualizaci měření teploty a relativní vlhkosti, která probíhá v reálném čase. V druhé je možné si vybrat vstup, na kterém má být proveden převod analogového signálu, umožňuje také povolení režimu *Override*. Aby nedošlo k nechtěným úpravám parametrů, jsou během aktivního měření ovládací prvky aplikace neaktivní. Pro vizualizaci většího množství dat přichozícího z A/D převodníku byl implementován časovač pomocí *Handle*, který periodicky obnovuje hodnoty v grafu. Poslední záložka je vytvořena pro nastavení frekvence a intervalu měření, pojmenování a exportu vybraných dat a k vyžádání kratšího, případně delšího intervalu spojení pro umožnění snížení energetických nároků pro BGM113.

V závěru byla aplikace úspěšně otestována a byla stanovena průměrná spotřeba BGM113 modulu při daných činnostech.

V návaznosti na tuto práci by bylo vhodné zpracovat energetické módy Bluetooth modulu a provést implementaci A/D převodníku, fungujícího během těchto módů. Aplikace i firmware jsou modulární, umožňují tak přidání nových funkcí, v případě této práce, připojení nových senzorů či rozšíření možností vizualizace dat. Vhodné by bylo zpracovat prohlížeč exportovaných dat, v této verzi totiž nelze prohlížet předchozí naměřená data přímo v aplikaci.

Literatura

- [1] HEYDON, Robin. *Bluetooth low energy: the developer's handbook*. Upper Saddle River, NJ: Prentice Hall, 2012. ISBN 013288836X.
- [2] SILICON LABS. *BGM113 Bluetooth Module Radio Board User's Guide*. Rev 2.00. Austin (Texas): Silicon Labs, 2016.
- [3] SILICON LABS. *I2C Humidity and Temperature Sensor: Si 7021 - A20*. Rev 1.2. Austin (Texas): Silicon Labs, 2016.
- [4] BLUETOOTH SPECIAL INTEREST GROUP. *Bluetooth: GATT Specifications* [online]. Kirkland (Washington): Bluetooth Special Interest Group, 2018. Dostupné z: <https://www.bluetooth.com/specifications/gatt>
- [5] MAHDIANI, Shadi, Vala JEYHANI, Mikko PELTOKANGAS a Antti VEIKKAJA. Is 50 Hz high enough ECG sampling frequency for accurate HRV analysis?. In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* [online]. IEEE, 2015, 2015, s. 5948-5951 DOI: 10.1109/EMBC.2015.7319746. ISBN 978-1-4244-9271-8.
- [6] CHOI, A a H SHIN. Photoplethysmography sampling frequency: pilot assessment of how low can we go to analyze pulse rate variability with reliability?. *Physiological Measurement* [online]. 2017, **38**(3), 586-600. DOI: 10.1088/1361-6579/aa5efa. ISSN 0967-3334.
- [7] SILICON LABS. *Bluetooth Software API Reference Manual*. Rev 2.8.0. Austin (Texas): Silicon Labs, 2018.
- [8] REN, Kai. Exploring Bluetooth 5 - How Fast Can It Be?. In: *Bluetooth: Bluetooth blog* [online]. Kirkland (Washington): Bluetooth Special Interest Group, 20 February 2017.
- [9] KARAMI, Kianoosh. Maximizing BLE Throughput on iOS and Android. In: *PunchThrough: Whitepapers* [online]. San Francisco (California): PunchThrough, 3 April 2016.
- [10] BLUETOOTH SPECIAL INTEREST GROUP. *Bluetooth Core Specification*. Vol 2: Core System Package. V 5.0. Kirkland (Washington): Bluetooth Special Interest Group, 2016.
- [11] SILICON LABS. *Bluetooth Smart Software Release Notes: Bluetooth Smart Software 2.4.0. Build 2328*. Austin (Texas): Silicon Labs, 2017.
- [12] SILICON LABS. *Silicon Labs Bluetooth ® C Application Developer's Guide*. Rev 1.1. Austin (Texas): Silicon Labs, 2018.
- [13] SILICON LABS. *AN0021: Analog to Digital Converter (ADC)*. Rev 1.11. Austin (Texas): Silicon Labs, 2016.
- [14] SILICON LABS. *BGM113 Blue Gecko Bluetooth ® Module Data Sheet*. Rev 1.00. Austin (Texas): Silicon Labs, 2016.
- [15] GOOGLE. *Android Developers: Docs* [online]. Mountain View (California): Google, 2018. Dostupné také z: <https://developer.android.com/docs/>

Seznam příloh

Příloha A. Zdrojové kódy firmware BGM113

Příloha B. Zdrojové kódy aplikace pro Android

Uvedené přílohy se nacházejí na přiloženém CD.